Tjedan 6: Vizualizacija podataka s ggplot2

Od brojeva do priča koje se vide

2025-03-22

Table of contents

```
library(tidyverse)
```

i Ishodi učenja

Nakon ovog predavanja moći ćete

1. Objasniti logiku gramatike grafike (grammar of graphics) i zašto je ggplot2 organiziran oko slojeva.
2. Identificirati tri obavezne komponente svakog ggplot2 grafa — podatke, estetike

(`aes()`) i geometriju (`geom_*()`) — te razumjeti njihovu ulogu.
3. Kreirati histograme i grafove gustoće za vizualizaciju distribucija jedne kontinuirane varijable.
4. Kreirati stupčaste grafove (`geom_bar()` i `geom_col()`) za prikaz kategoričkih varijabli i njihovih frekvencija ili sažetaka.
5. Kreirati boxplotove i violin grafove za usporedbu distribucija jedne varijable između grupa.
6. Kreirati točkaste grafove (scatterplots) za vizualizaciju odnosa između dviju kontinuiranih varijabli.
7. Koristiti estetike boje, ispune, oblika i veličine za kodiranje dodatnih varijabli u grafu.
8. Prilagoditi oznake osi, naslove i podnaslove pomoću `labs()`.

1 Zašto je vizualizacija važna

Prošli tjedan naučili smo izračunati prosjek, medijan, standardnu devijaciju i korelaciju. To su korisni brojevi, ali sami po sebi rijetko govore cijelu priču. Anscombe je 1973. konstruirao četiri dataseta koji imaju identičan prosjek (7.5 za x, identičan za y), identičnu standardnu devijaciju, identičnu korelaciju (0.816) i identičnu regresijsku liniju. A kad ih nacrtate, vidite četiri potpuno različita uzorka. Jedan je linearan, drugi je zakrivljen, treći ima jedan outlier koji povlači liniju, četvrti ima grupirane točke s jednim ekstremom. Bez vizualizacije, sva četiri izgledaju jednako.

Ista logika vrijedi za svaku analizu koju ćete raditi kao komunikolozi. Recimo da vam netko kaže "prosječno vrijeme čitanja članaka na našem portalu je 83 sekunde". Zvuči informativno. Ali to vam ne govori je li distribucija simetrična ili iskrivljena. Možda većina čitatelja provede 30 sekundi, a nekolicina koja čita detaljno diže prosjek. Ili možda postoje dva jasna klastera — oni koji odmah odu (bounce) i oni koji čitaju do kraja. Histogram bi to pokazao u sekundi. Broj sam po sebi ne može.

U ovom tjednu učimo ggplot2, paket za vizualizaciju koji je dio tidyverse ekosustava. ggplot2 nije samo alat za crtanje grafova. On implementira konzistentnu logiku (gramatiku grafike) koja vam omogućuje da razmišljate o vizualizaciji na strukturiran način. Kad jednom shvatite tu logiku, moći ćete kreirati bilo koji graf od istih temeljnih komponenti.

---

2 Naši podaci: angažman čitatelja na portalima

Koristit ćemo simulirani dataset koji sadrži podatke o 1000 članaka objavljenih na hrvatskim informativnim portalima. Za svaki članak imamo informacije o izvoru, kategoriji, stilu naslova, formatu, broju riječi, vremenu provedenom na stranici, broju dijeljenja, komentara, dubini scrollanja i drugim metrikama angažmana.

```r
clanci <- read_csv("../resources/datasets/article_engagement.csv")
glimpse(clanci)
```

```
Rows: 1,000
Columns: 16
$ article_id     <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ~
$ source         <chr> "Večernji.hr", "Index.hr", "Index.hr", "Index.hr", "24s~
$ category       <chr> "Politika", "Tehnologija", "Politika", "Sport", "Politi~
$ headline_style <chr> "informativni", "senzacionalistički", "narativni", "inf~
$ format         <chr> "tekst+slika", "tekst+slika", "tekst+video", "tekst+sli~
$ word_count     <dbl> 624, 191, 763, 249, 1117, 766, 661, 795, 394, 177, 466,~
$ has_image      <lgl> TRUE, TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, TRUE, FALSE,~
$ has_video      <lgl> FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
$ publish_hour   <dbl> 20, 0, 14, 7, 11, 19, 8, 22, 22, 20, 18, 16, 4, 16, 7, ~
$ day_of_week    <chr> "ponedjeljak", "subota", "nedjelja", "utorak", "nedjelj~
$ time_on_page   <dbl> 88, 22, 191, 30, 113, 100, 86, 208, 21, 23, 55, 102, 44~
$ shares         <dbl> 0, 1, 7, 0, 11, 9, 0, 2, 1, 0, 0, 1, 1, 0, 1, 5, 10, 3,~
$ comments       <dbl> 0, 5, 5, 0, 14, 3, 0, 1, 0, 0, 0, 0, 1, 1, 3, 7, 1, 0, ~
$ scroll_depth   <dbl> 57, 33, 88, 48, 30, 34, 54, 75, 21, 100, 5, 86, 53, 81,~
$ bounce         <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE,~
$ return_visit   <lgl> TRUE, TRUE, TRUE, FALSE, FALSE, TRUE, FALSE, FALSE, FAL~
```

Dataset ima 1000 redova i 16 stupaca. Već na prvi pogled vidimo mješavinu kontinuiranih (word_count, time_on_page, scroll_depth), kategoričkih (source, category, headline_style) i logičkih (has_image, bounce, return_visit) varijabli. Ova raznolikost je idealna za učenje vizualizacije jer svaki tip varijable traži drugačiji tip grafa.

Pogledajmo osnovne karakteristike.

```r
clanci |>
  count(source, sort = TRUE)
```

```
# A tibble: 7 x 2
  source          n
  <chr>       <int>
1 Index.hr      254
2 24sata.hr     203
3 Jutarnji.hr   175
4 Večernji.hr   157
5 N1info.hr      92
6 Telegram.hr    69
7 tportal.hr     50
```

```r
clanci |>
  count(category, sort = TRUE)
```

```
# A tibble: 7 x 2
  category         n
  <chr>        <int>
1 Politika       224
2 Lifestyle      198
3 Sport          186
4 Tehnologija    138
5 Kultura        118
6 Znanost         82
7 Crna kronika    54
```

```r
clanci |>
  summarise(
    prosjek_vrijeme = round(mean(time_on_page), 1),
    prosjek_rijeci = round(mean(word_count), 0),
    prosjek_dijeljenja = round(mean(shares), 1)
  )
```

```
# A tibble: 1 x 3
  prosjek_vrijeme prosjek_rijeci prosjek_dijeljenja
            <dbl>          <dbl>               <dbl>
1            83.4            511                 4.5
```

Sad kad znamo s čime radimo, krenimo graditi grafove.

---

3 Gramatika grafike: kako ggplot2 razmišlja

Paket ggplot2 temelji se na knjizi *The Grammar of Graphics* (Wilkinson, 2005), koja opisuje vizualizaciju podataka kao sustav komponenti koje se slažu u slojeve. Ideja je da svaki graf, koliko god bio složen, nastaje kombinacijom istih temeljnih elemenata.

Tri elementa su obavezna za svaki ggplot2 graf.

Podaci (data) — tibble koji sadrži varijable koje želite prikazati.

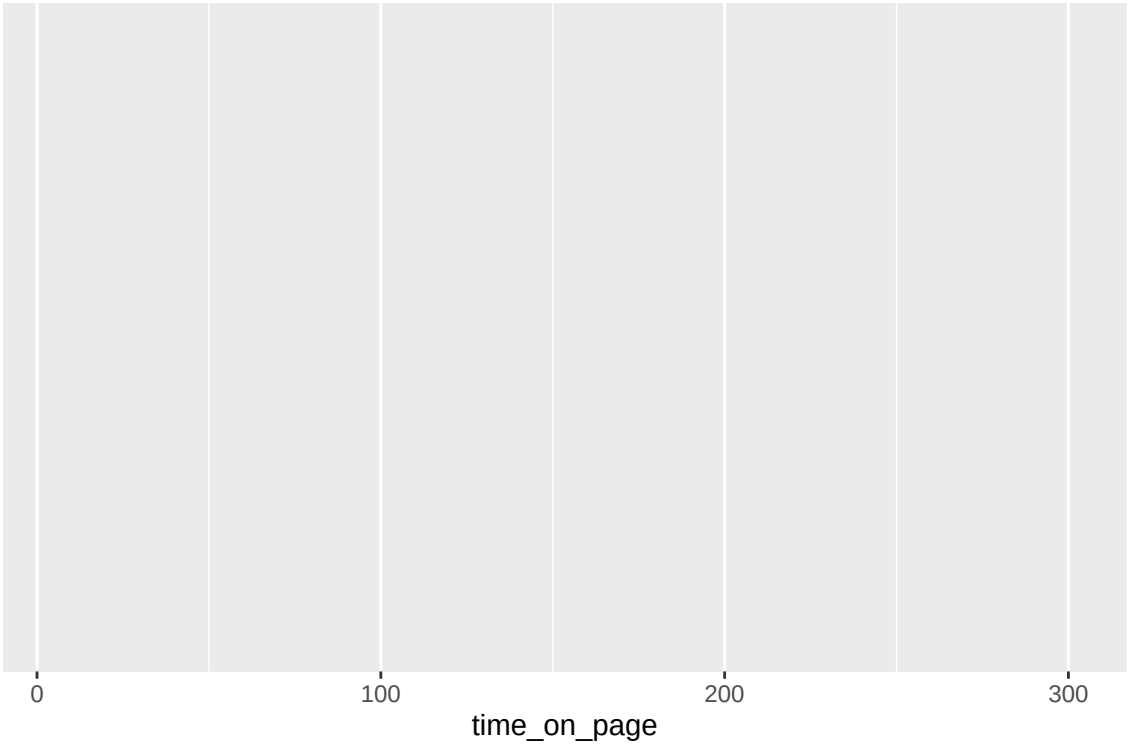
Estetike (aesthetics, `aes()`) — mapiranja varijabli na vizualne dimenzije grafa. Na primjer, varijabla `time_on_page` ide na x os, `shares` na y os, `category` određuje boju.

Geometrija (geometry, `geom_*()`) — oblik kojim se podaci prikazuju, kao što su točke za scatterplot (`geom_point()`), stupci za bar chart (`geom_bar()`) ili linije za linijski graf (`geom_line()`).

Osim ova tri, graf može imati i dodatne slojeve poput statističkih transformacija (stat), prilagodbi skala (scale), podjele u panele (facet), koordinatnog sustava (coord) i vizualne teme (theme). Svaki od ovih elemenata se dodaje operatorom `+`.

Pogledajmo najjednostavniji mogući ggplot2 kod.

```
ggplot(data = clanci, mapping = aes(x = time_on_page))
```



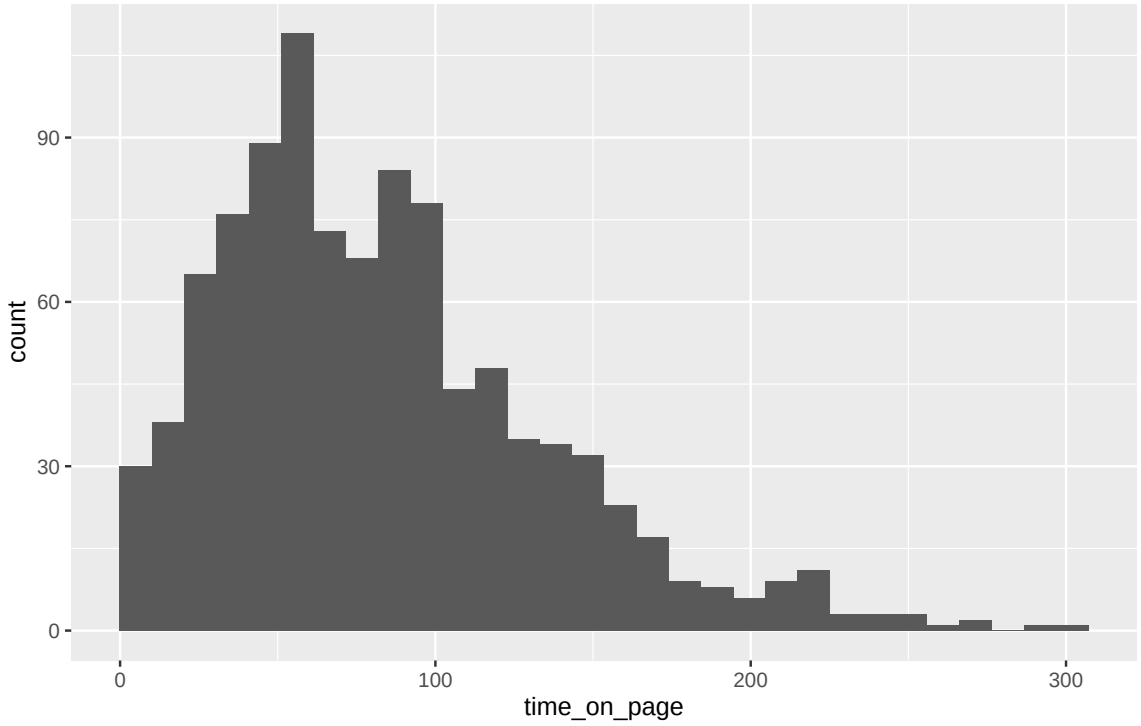
Ovaj kod kreira prazan graf. Definirali smo podatke (`clanci`) i jednu estetiku (varijabla `time_on_page` na x osi), ali nismo rekli ggplotu KAKO da prikaže te podatke (nismo dodali geometriju). Rezultat je prazan koordinatni sustav s ispravno postavljenom x osi. Ggplot zna raspon varijable i pripremio je platno, ali čeka da mu kažemo što da nacrta.

---

4 Histogrami: distribucija jedne varijable

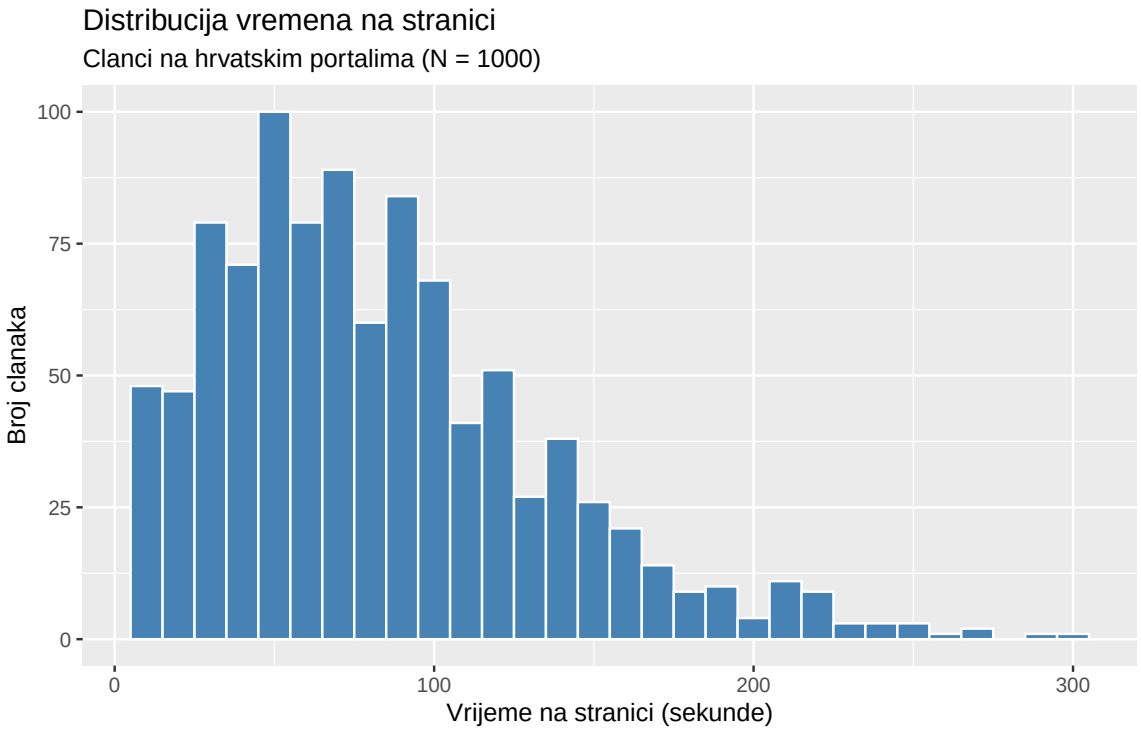
Histogram je najvažniji graf za razumijevanje distribucije jedne kontinuirane varijable. Dijeli raspon vrijednosti u jednake intervale (binove) i prikazuje koliko opažanja pada u svaki interval.

```r
ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram()
```



Ggplot nas upozorava da koristi 30 binova i sugerira da eksperimentiramo s `bins` ili `binwidth` argumentom. Upozorenje je korisno jer broj binova značajno utječe na to što vidimo. S premalo binova gubimo detalje, s previše binova graf postaje neuredan.

```r
ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram(binwidth = 10, fill = "steelblue", color = "white") +
  labs(
    title = "Distribucija vremena na stranici",
    subtitle = "Članci na hrvatskim portalima (N = 1000)",
    x = "Vrijeme na stranici (sekunde)",
    y = "Broj članaka"
  )
```



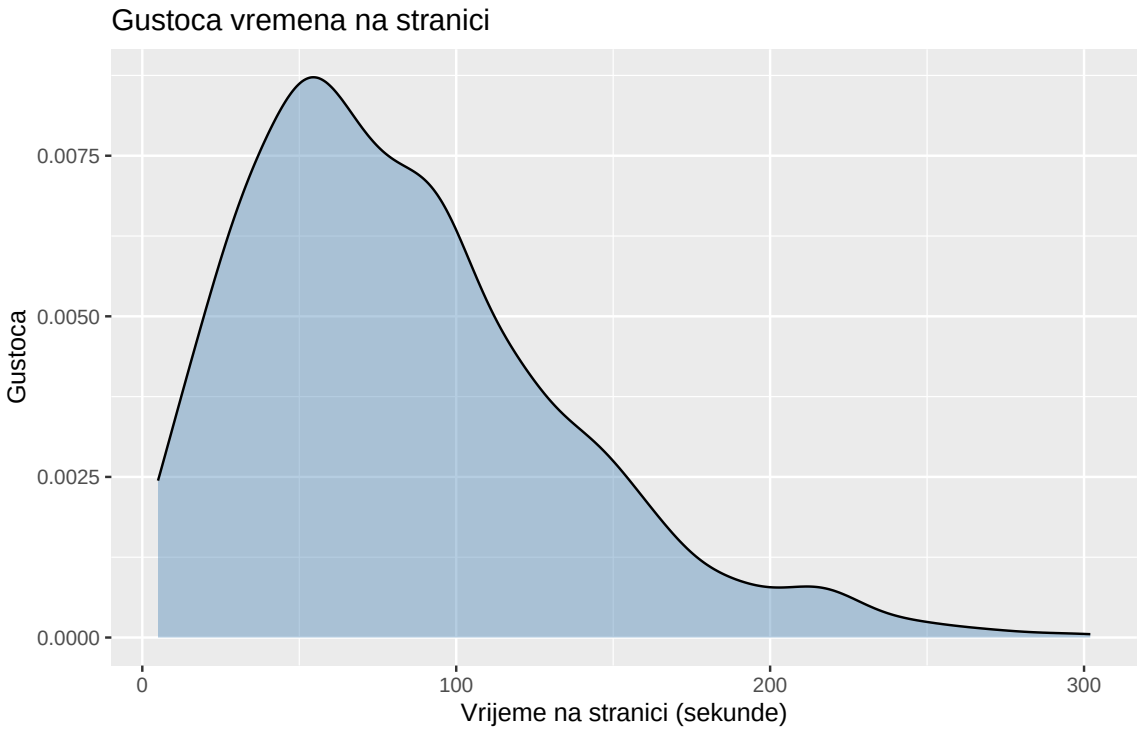
Ovdje smo napravili nekoliko poboljšanja. Argument `binwidth = 10` postavlja širinu svakog bina na 10 sekundi, što daje jasniju sliku od defaultnih 30 binova. `fill = "steelblue"` boja ispunu stupaca, a `color = "white"` crta bijeli rub između stupaca za bolju čitljivost. Funkcija `labs()` dodaje naslov, podnaslov i oznake osi.

Distribucija je desno iskrivljena (pozitivan skew), što je tipično za metriku angažmana. Većina članaka ima relativno kratko vrijeme čitanja, ali postoji dugačak rep članaka s izuzetno dugim vremenom čitanja.

4.1 Graf gustoće (density plot)

Alternativa histogramu je graf gustoće koji prikazuje procijenjenu krivulju gustoće vjerojatnosti. Prednost je što ne ovisi o odabiru binova i daje glatku krivulju.

```
ggplot(clanci, aes(x = time_on_page)) +
  geom_density(fill = "steelblue", alpha = 0.4) +
  labs(
    title = "Gustoća vremena na stranici",
    x = "Vrijeme na stranici (sekunde)",
    y = "Gustoća"
  )
```

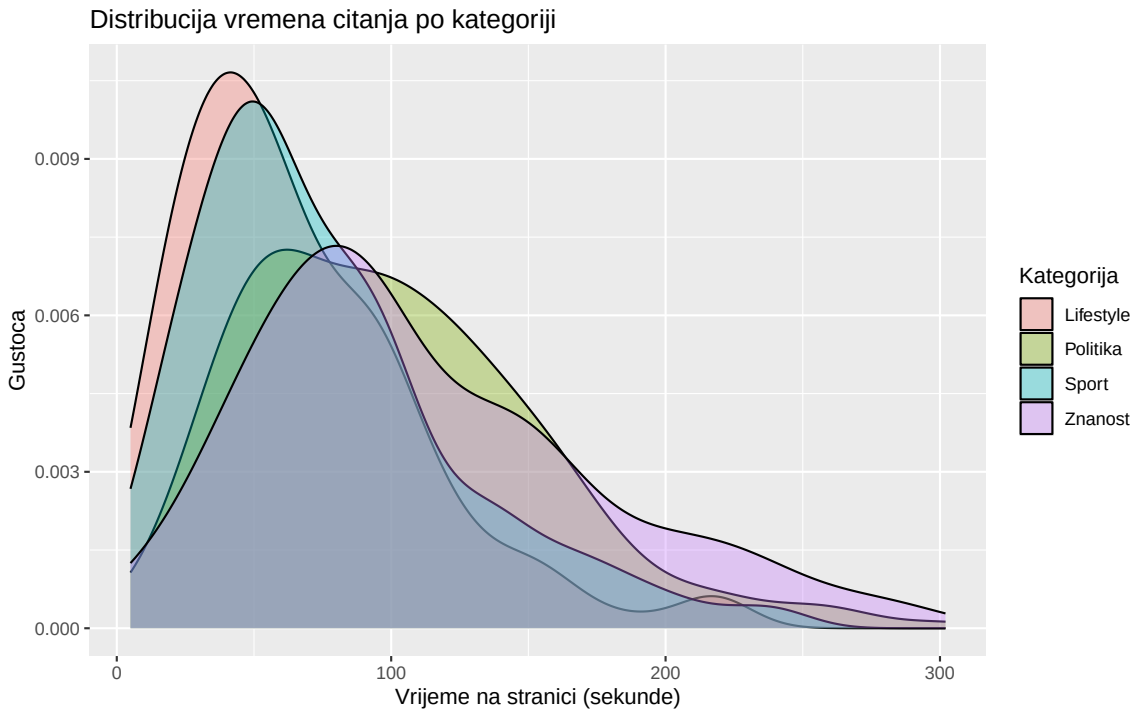


Argument `alpha = 0.4` kontrolira transparentnost ispune (0 je potpuno prozirno, 1 potpuno neprozirno). Transparentnost je osobito korisna kad preklapate više distribucija.

4.2 Usporedba distribucija s density plotom

Recimo da želimo usporediti distribuciju vremena čitanja između različitih kategorija članaka. Histogram bi bio nepregledan s pet ili više preklopljenih boja, ali graf gustoće radi dobro.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost", "Lifestyle")) |>
  ggplot(aes(x = time_on_page, fill = category)) +
  geom_density(alpha = 0.35) +
  labs(
    title = "Distribucija vremena čitanja po kategoriji",
    x = "Vrijeme na stranici (sekunde)",
    y = "Gustoća",
    fill = "Kategorija"
  )
```



Primijetite novu estetiku `fill = category` unutar `aes()`, koja govori ggplotu da koristi različitu boju ispune za svaku kategoriju. Kad je estetika mapirana na varijablu (unutar `aes()`), ggplot automatski kreira legendu.

Vidimo da znanstveni članci imaju širu distribuciju pomaknuto udesno (duže čitanje), dok su lifestyle članci koncentrirani na kraćem kraju. Politički članci su negdje između. Ovo ima smisla jer znanstveni članci tendiraju biti duži i zahtijevaju više pozornosti.

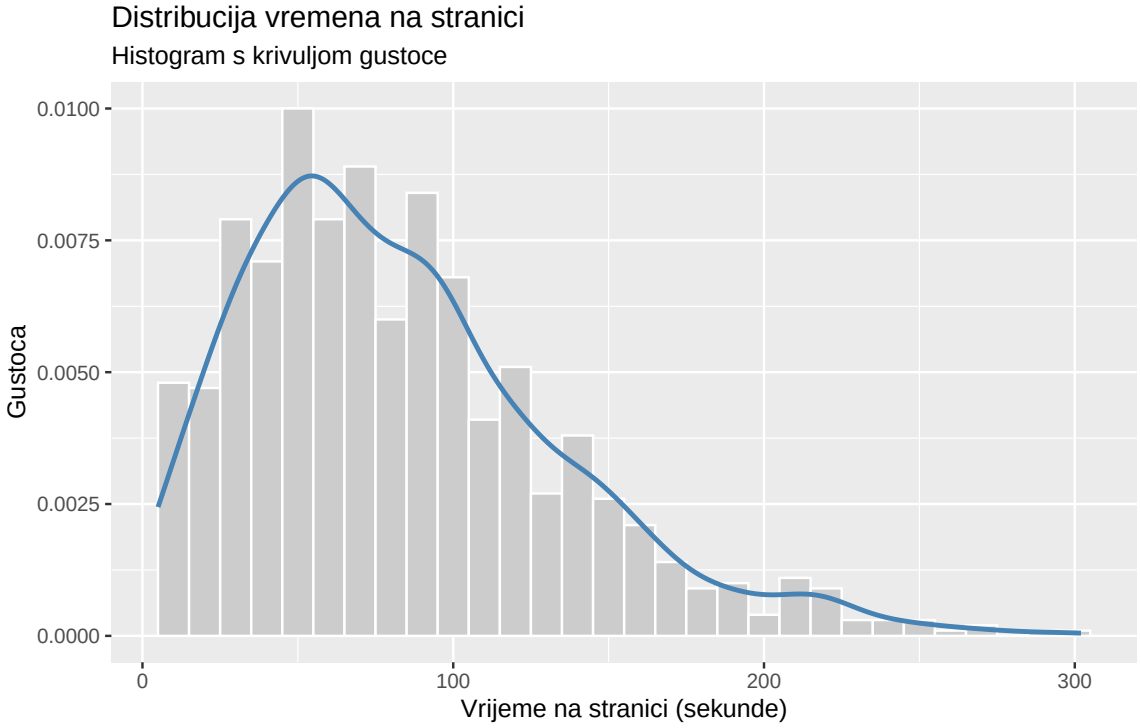
> **Praktični savjet**
>
> Kad prikazujete distribucije više grupa, density plot je gotovo uvijek bolji izbor od preklopljenih histograma. Histogrami se preklapaju i zaklanjaju jedni druge, dok su density krivulje s transparentnošću (`alpha < 0.5`) lako čitljive i za četiri ili pet grupa. Za više od pet grupa, razmislite o facetiranju (koje ćemo naučiti u drugom dijelu).

4.3 Histogram i density zajedno

Ponekad je korisno nacrtati oboje na istom grafu. Za to moramo histogramu reći da na y osi prikaže gustoću umjesto broja opažanja, kako bi skale bile usporedive.

```
ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram(
    aes(y = after_stat(density)),
    binwidth = 10,
```

```
    fill = "grey80",
    color = "white"
  ) +
  geom_density(color = "steelblue", linewidth = 1) +
  labs(
    title = "Distribucija vremena na stranici",
    subtitle = "Histogram s krivuljom gustoće",
    x = "Vrijeme na stranici (sekunde)",
    y = "Gustoća"
  )
```



Ključan detalj je `aes(y = after_stat(density))` unutar `geom_histogram()`. Ovo govori ggplotu da na y osi prikaže gustoću (proporciju) umjesto apsolutnog broja, čime histogram i krivulja gustoće postaju usporedivi.

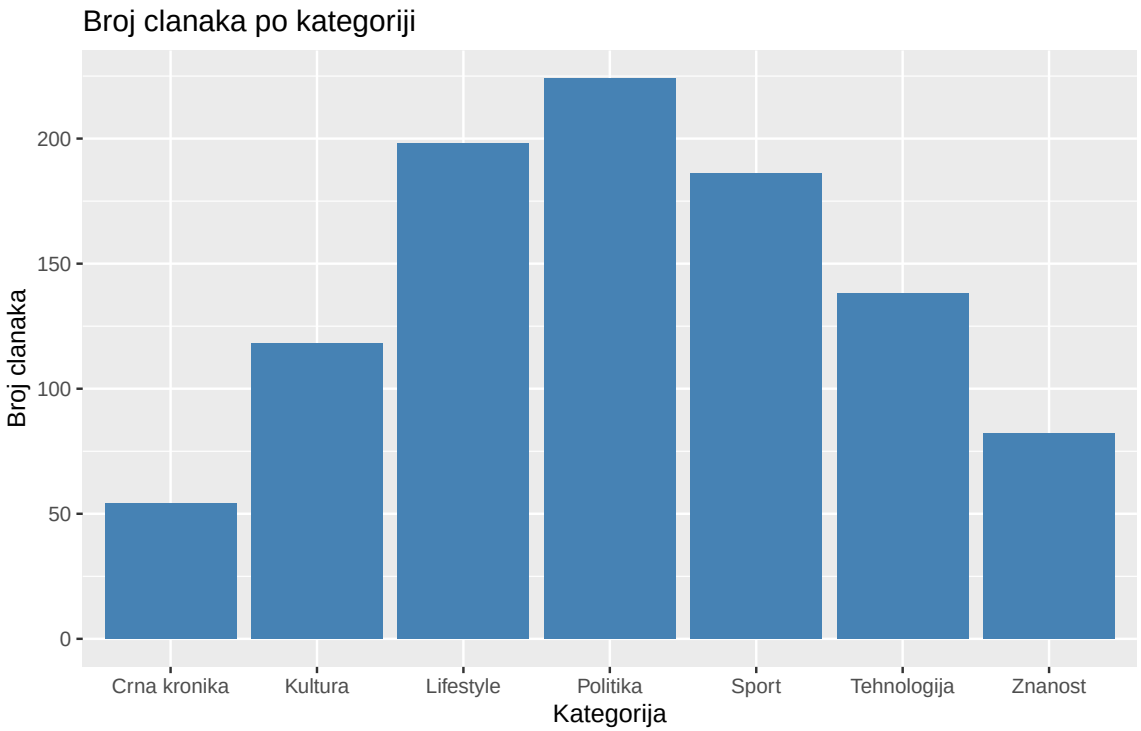
---

5 Stupčasti grafovi: kategoričke varijable

Stupčasti grafovi (bar charts) prikazuju frekvencije ili sažetke za kategoričke varijable. U ggplot2 postoje dvije varijante — `geom_bar()` koja sama broji opažanja i `geom_col()` koja prikazuje unaprijed izračunate vrijednosti.

5.1 geom_bar(): automatsko prebrojavanje

```
ggplot(clanci, aes(x = category)) +
  geom_bar(fill = "steelblue") +
  labs(
    title = "Broj članaka po kategoriji",
    x = "Kategorija",
    y = "Broj članaka"
  )
```



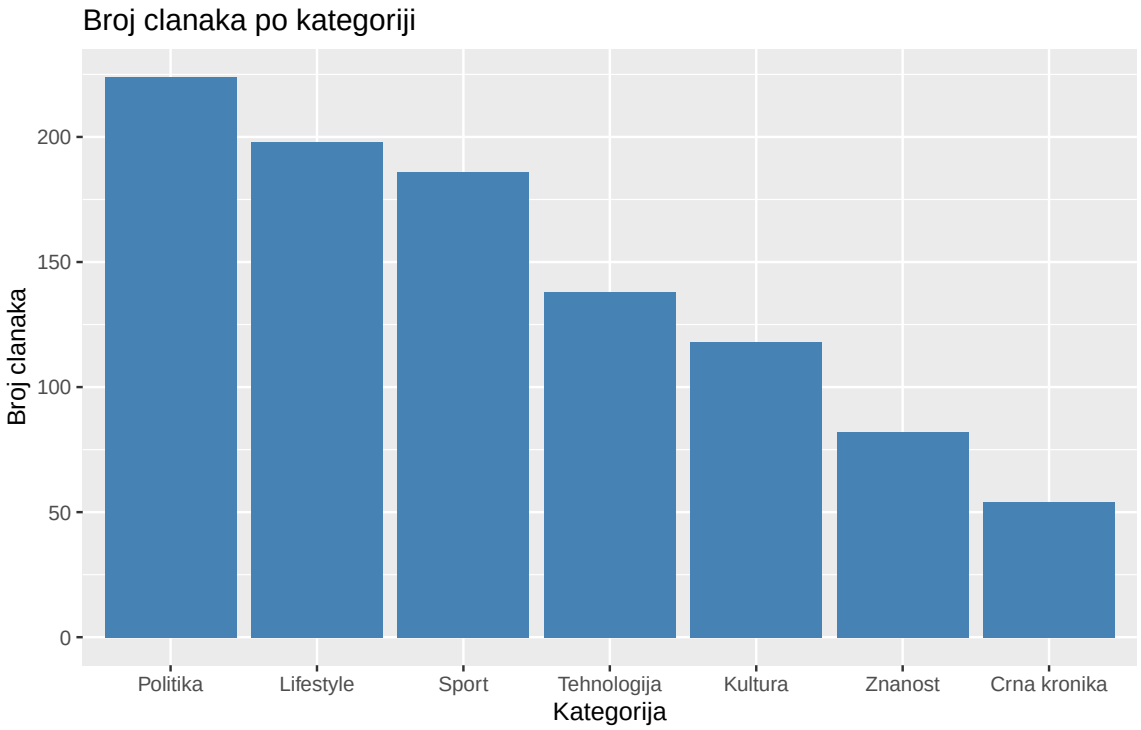
`geom_bar()` automatski broji koliko redova (članaka) pripada svakoj kategoriji i prikazuje rezultat kao stupac. Ovo je ekvivalent pozivanja `count()` na podatke, ali vizualno.

5.2 Sortiranje stupaca po veličini

Abecedni redoslijed kategorija rijetko je informativan. Bolje je sortirati stupce po veličini pomoću `fct_infreq()` iz paketa forcats (dio tidyverse).

```
ggplot(clanci, aes(x = fct_infreq(category))) +
  geom_bar(fill = "steelblue") +
  labs(
    title = "Broj članaka po kategoriji",
```

```
    x = "Kategorija",
    y = "Broj članaka"
  )
```

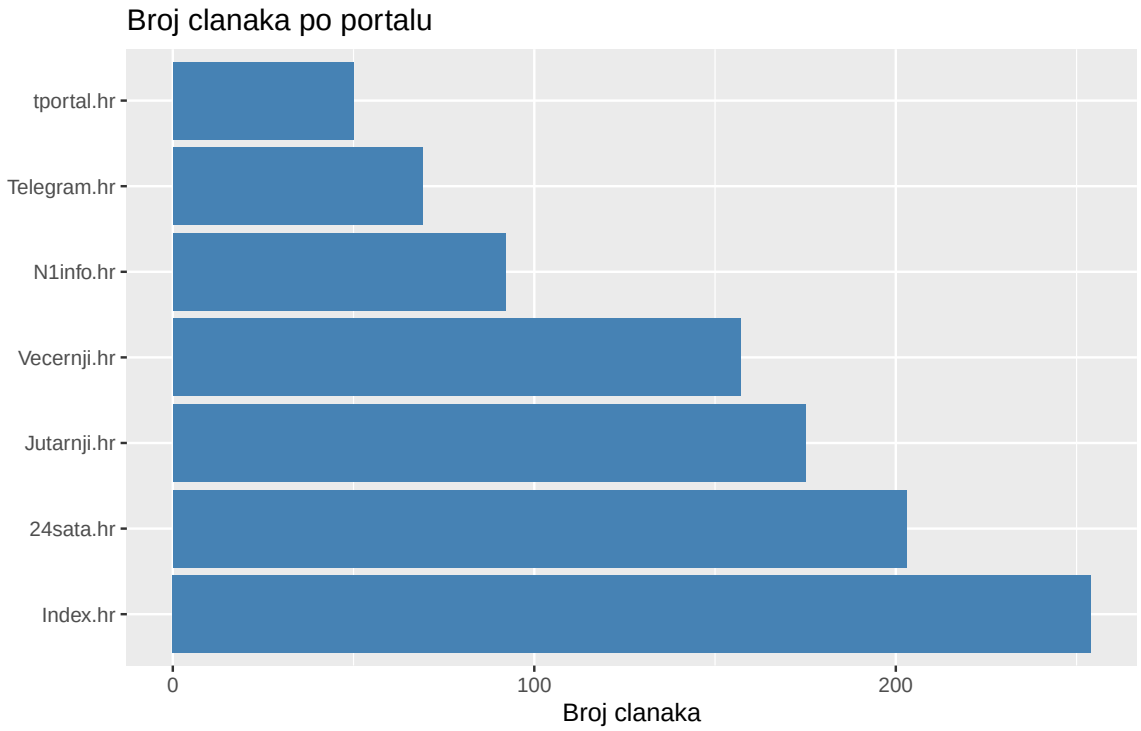


Sad je odmah vidljivo da Politika ima najviše članaka, a Crna kronika najmanje. Funkcija `fct_infreq()` slaže kategorije od najčešće prema najrjeđoj. Za obrnuti redoslijed, omotajte u `fct_rev()` — `fct_rev(fct_infreq(category))`.

5.3 Horizontalni stupčasti graf

Za kategorije s dugačkim imenima, horizontalni graf je čitljiviji.

```
ggplot(clanci, aes(y = fct_infreq(source))) +
  geom_bar(fill = "steelblue") +
  labs(
    title = "Broj članaka po portalu",
    x = "Broj članaka",
    y = NULL
  )
```

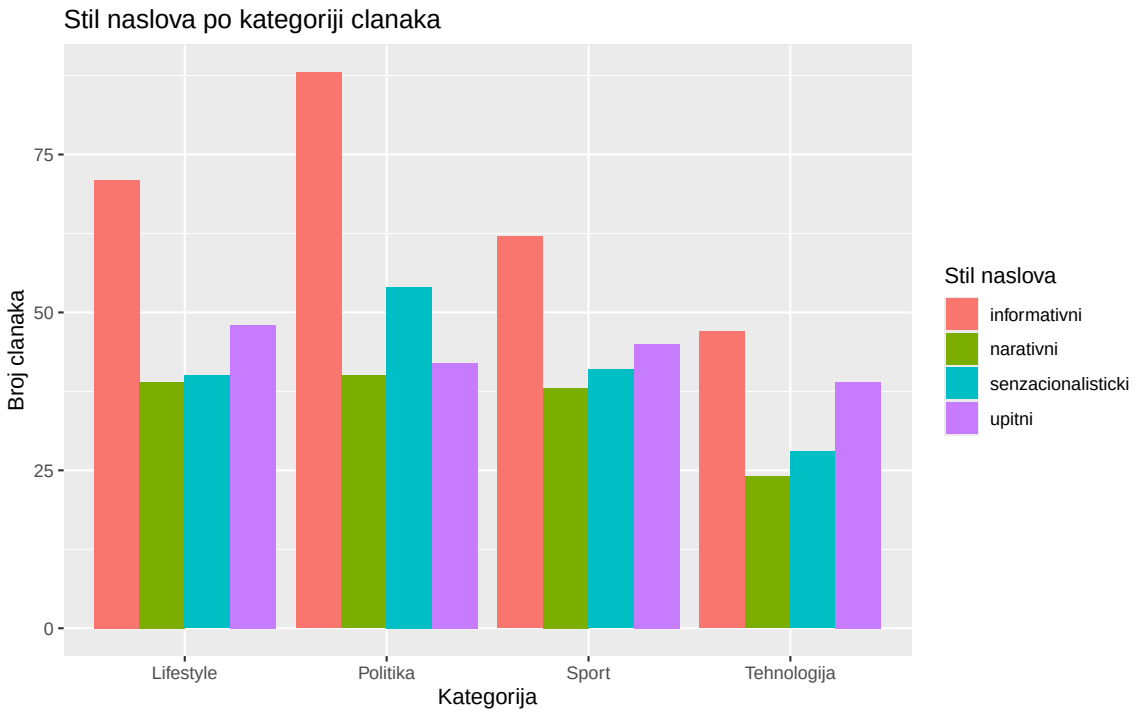


Trik je jednostavan — umjesto `x` koristite `y` u `aes()`, i ggplot automatski crta horizontalne stupce. Postavili smo `y = NULL` u `labs()` da uklonimo nepotrebnu oznaku osi jer su imena portala samorazumljiva.

5.4 Grupirani i složeni stupčasti grafovi

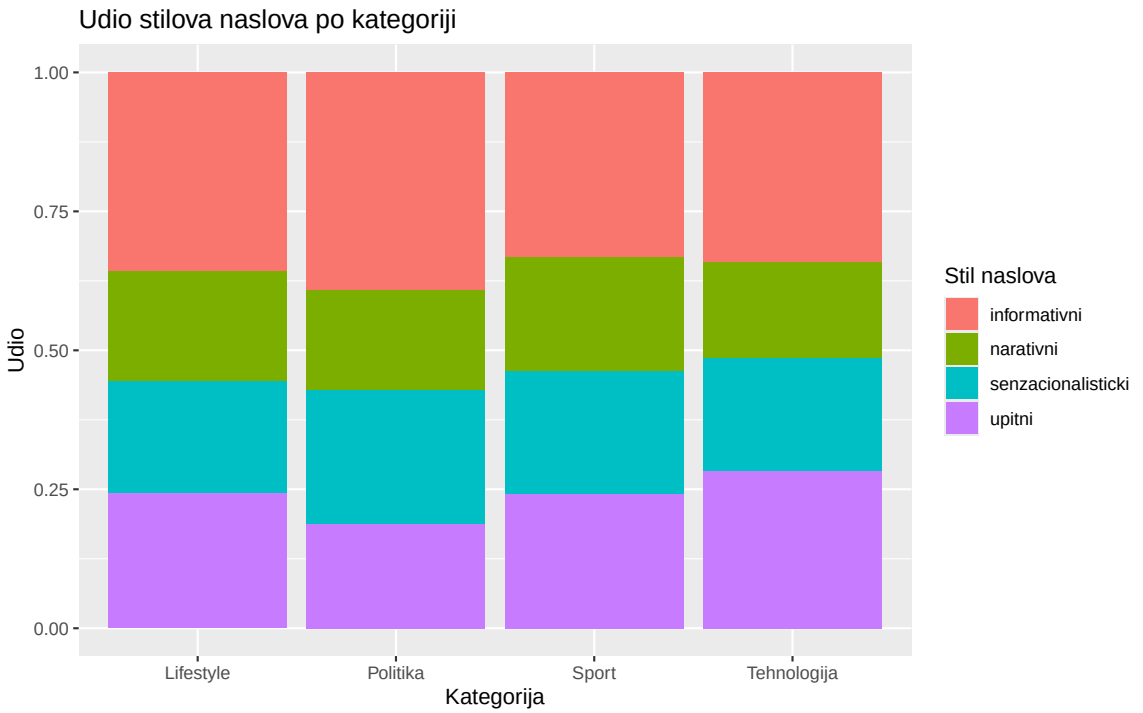
Kad želimo prikazati odnos između dviju kategoričkih varijabli, koristimo boju za drugu varijablu.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Tehnologija", "Lifestyle")) |>
  ggplot(aes(x = category, fill = headline_style)) +
  geom_bar(position = "dodge") +
  labs(
    title = "Stil naslova po kategoriji članaka",
    x = "Kategorija",
    y = "Broj članaka",
    fill = "Stil naslova"
  )
```



Argument `position = "dodge"` postavlja stupce jedne do drugih umjesto da ih slaže. Alternativa je `position = "fill"` koji prikazuje proporcije umjesto apsolutnih brojeva.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Tehnologija", "Lifestyle")) |>
  ggplot(aes(x = category, fill = headline_style)) +
  geom_bar(position = "fill") +
  labs(
    title = "Udio stilova naslova po kategoriji",
    x = "Kategorija",
    y = "Udio",
    fill = "Stil naslova"
  )
```

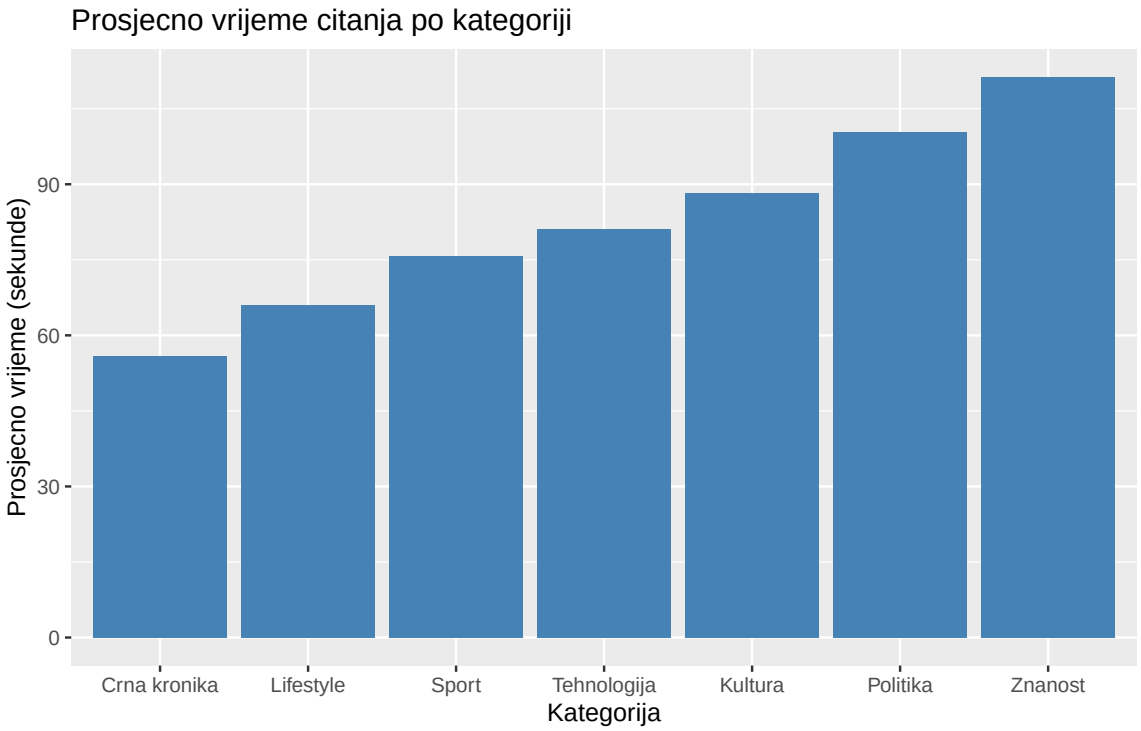


Ovaj graf otkriva zanimljive obrasce. Proporcija senzacionalističkih naslova se razlikuje po kategorijama. Ovo je vizualna verzija tablice unakrsnih frekvencija (contingency table) i koristit ćete ju kad budemo radili hi-kvadrat testove u tjednu 11.

5.5 geom_col(): vlastiti sažeci

Kad ste već izračunali sažetke (prosjeke, medijane, postotke) pomoću `summarise()`, koristite `geom_col()` koji očekuje gotove y vrijednosti.

```
clanci |>
  group_by(category) |>
  summarise(prosjek_vrijeme = mean(time_on_page), .groups = "drop") |>
  mutate(category = fct_reorder(category, prosjek_vrijeme)) |>
  ggplot(aes(x = category, y = prosjek_vrijeme)) +
  geom_col(fill = "steelblue") +
  labs(
    title = "Prosječno vrijeme čitanja po kategoriji",
    x = "Kategorija",
    y = "Prosječno vrijeme (sekunde)"
  )
```



Ovdje smo najprije izračunali prosjeke, zatim koristili `fct_reorder()` da sortiramo kategorije po prosječnom vremenu (ne po frekvenciji kao `fct_infreq()`), i onda prikazali te prosjeke s `geom_col()`.

Razlika između `geom_bar()` i `geom_col()` je ključna. `geom_bar()` sam broji retke i ne treba y estetiku. `geom_col()` prikazuje y vrijednosti koje ste vi pripremili. Koristite `geom_bar()` za frekvencije, `geom_col()` za sve ostalo (prosjeke, medijane, postotke, bilo kakve prethodno izračunate sažetke).

> **!** Važna napomena
>
> Stupčasti graf prosjeka skriva distribuciju podataka. Kad vidite samo stupac visine 83, ne znate je li to zato što su svi oko 83 ili zato što su pola ljudi na 10 i pola na 156. Za usporedbu distribucija između grupa, boxplot ili violin plot su gotovo uvijek bolji izbor. Stupčaste grafove prosjeka koristite samo kad je publici dovoljna informacija o prosjecima (na primjer, u izvještaju za klijenta koji ne želi vidjeti boxplotove).

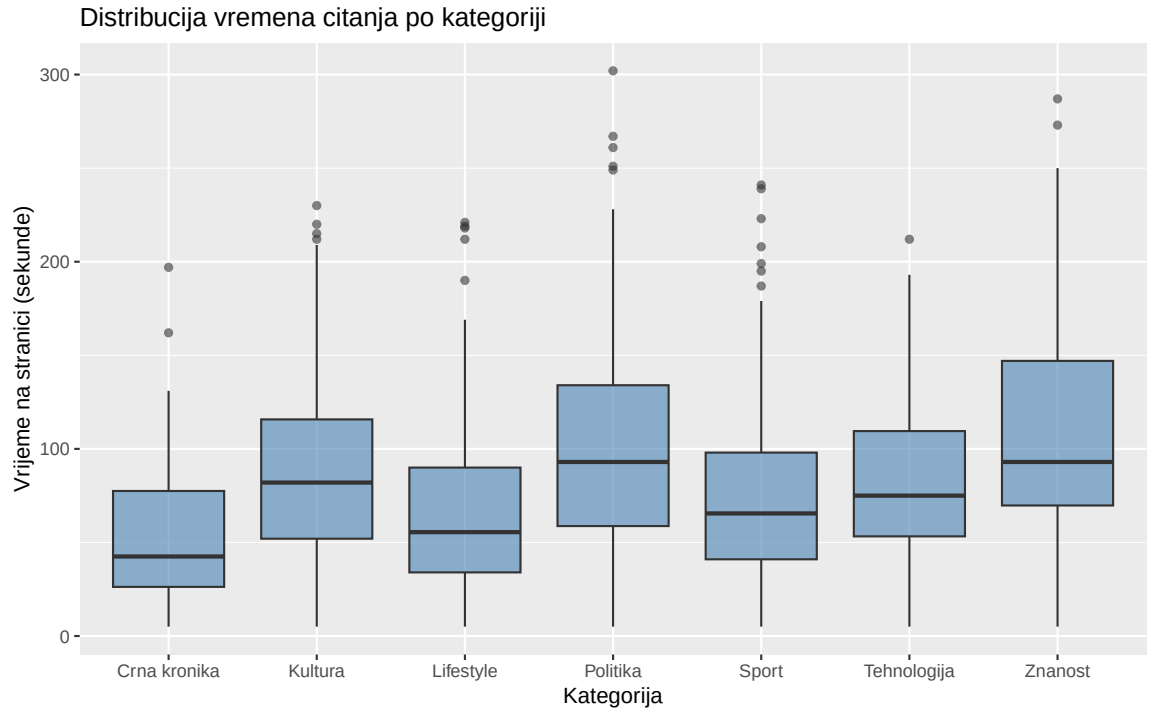
---

6 Boxplot: usporedba distribucija između grupa

Boxplot (dijagram pravokutnika) prikazuje pet ključnih brojeva distribucije — minimum, prvi kvartil (Q1), medijan, treći kvartil (Q3) i maksimum. Također identificira potencijalne

outliere. Za usporedbu distribucija između grupa, boxplot je jedan od najkorisnijih grafova.

```
ggplot(clanci, aes(x = category, y = time_on_page)) +
  geom_boxplot(fill = "steelblue", alpha = 0.6) +
  labs(
    title = "Distribucija vremena čitanja po kategoriji",
    x = "Kategorija",
    y = "Vrijeme na stranici (sekunde)"
  )
```



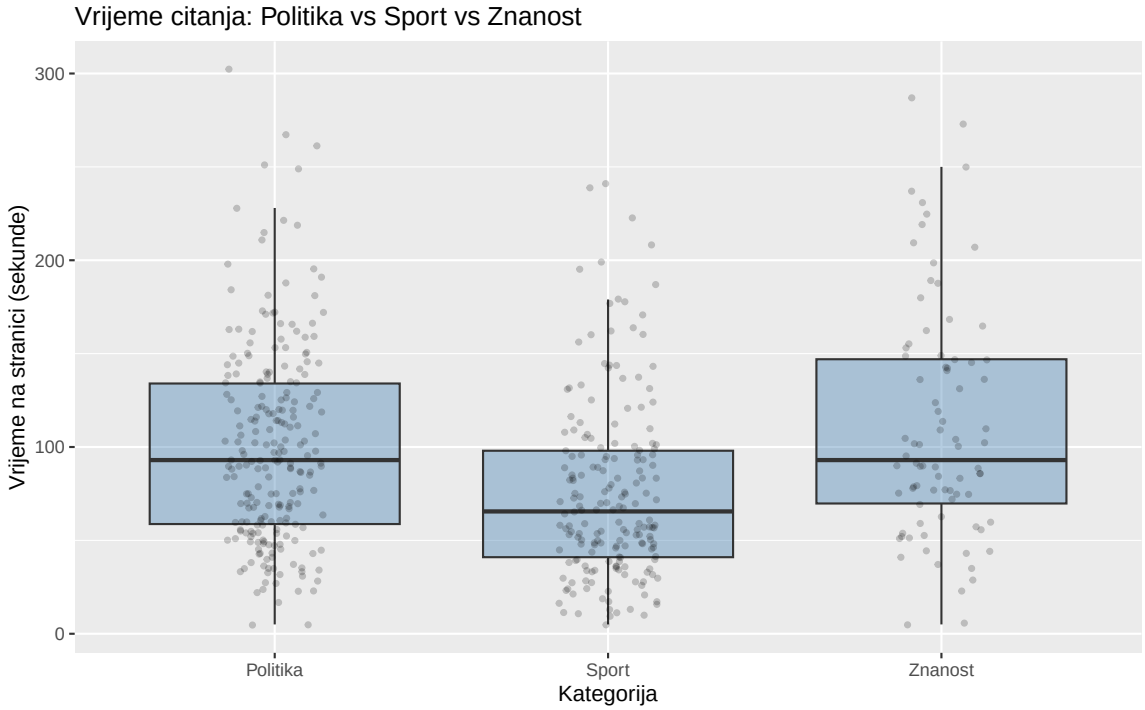
Čitanje boxplota ide na sljedeći način. Deblja crta unutar pravokutnika je medijan, donji rub pravokutnika je Q1 (25. percentil), a gornji rub je Q3 (75. percentil). Visina pravokutnika je interkvartilni raspon (IQR = Q3 − Q1), koji obuhvaća srednjih 50% podataka. Linije (whiskers) se protežu do najudaljenije točke koja je unutar 1.5 × IQR od ruba pravokutnika, dok su točke izvan toga potencijalni outlieri.

Iz ovog grafa jasno vidimo da znanstveni članci imaju ne samo viši medijan vremena čitanja nego i veću varijabilnost. Lifestyle članci su koncentrirani na nižim vrijednostima. Svaka kategorija ima outliere na desnoj strani, što je očekivano za metriku angažmana.

6.1 Boxplot s točkama

Boxplot sažima distribuciju u pet brojeva, pa neke informacije gubi. Dodavanje pojedinačnih točaka vraća taj kontekst.

```r
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost")) |>
  ggplot(aes(x = category, y = time_on_page)) +
  geom_boxplot(fill = "steelblue", alpha = 0.4, outlier.shape = NA) +
  geom_jitter(width = 0.15, alpha = 0.2, size = 1) +
  labs(
    title = "Vrijeme čitanja: Politika vs Sport vs Znanost",
    x = "Kategorija",
    y = "Vrijeme na stranici (sekunde)"
  )
```



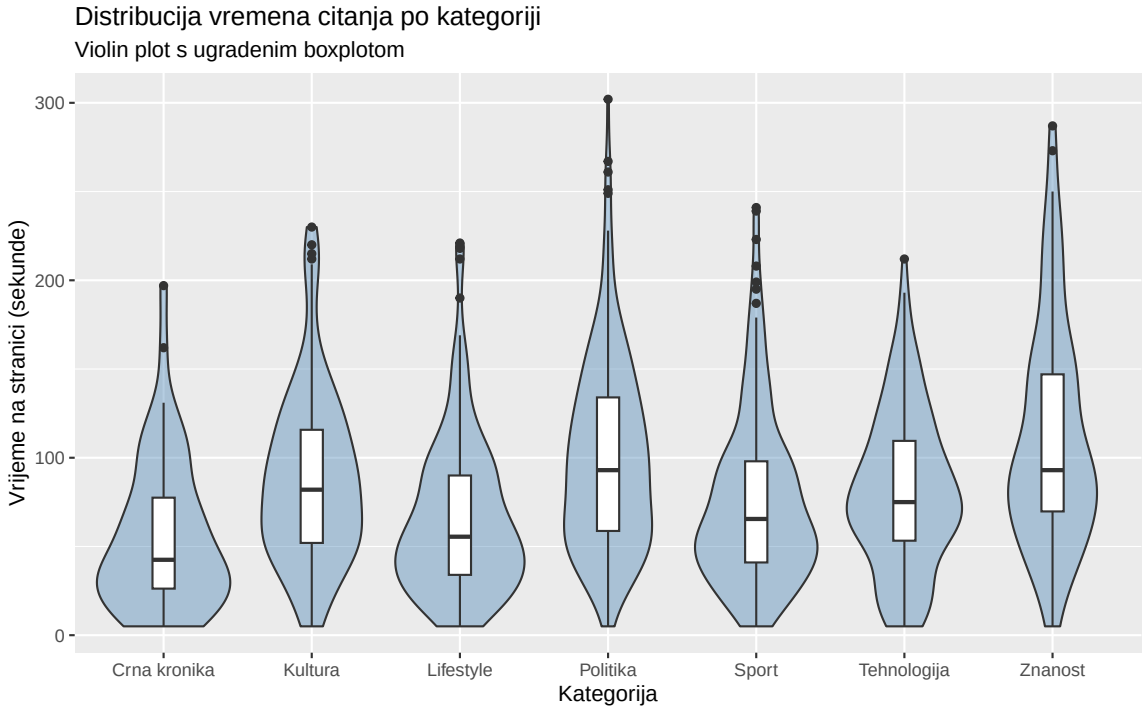
`geom_jitter()` dodaje točke s malim nasumičnim pomakom po horizontali (`width = 0.15`) da se ne preklapaju, a `alpha = 0.2` čini točke poluprozirnim kako bismo vidjeli gustoću. `outlier.shape = NA` u boxplotu isključuje prikaz outliera jer bi se inače udvostručili s jitter točkama.

Ovaj kombinirani prikaz daje kompletnu sliku: boxplot za sažetak distribucije i točke za uvid u stvarne podatke.

6.2 Violin plot: oblik distribucije

Violin plot je varijanta boxplota koja prikazuje oblik distribucije pomoću zrcaljene krivulje gustoće. Tamo gdje je graf širi, ima više podataka.

```r
ggplot(clanci, aes(x = category, y = time_on_page)) +
  geom_violin(fill = "steelblue", alpha = 0.4) +
  geom_boxplot(width = 0.15, fill = "white") +
  labs(
    title = "Distribucija vremena čitanja po kategoriji",
    subtitle = "Violin plot s ugrađenim boxplotom",
    x = "Kategorija",
    y = "Vrijeme na stranici (sekunde)"
  )
```



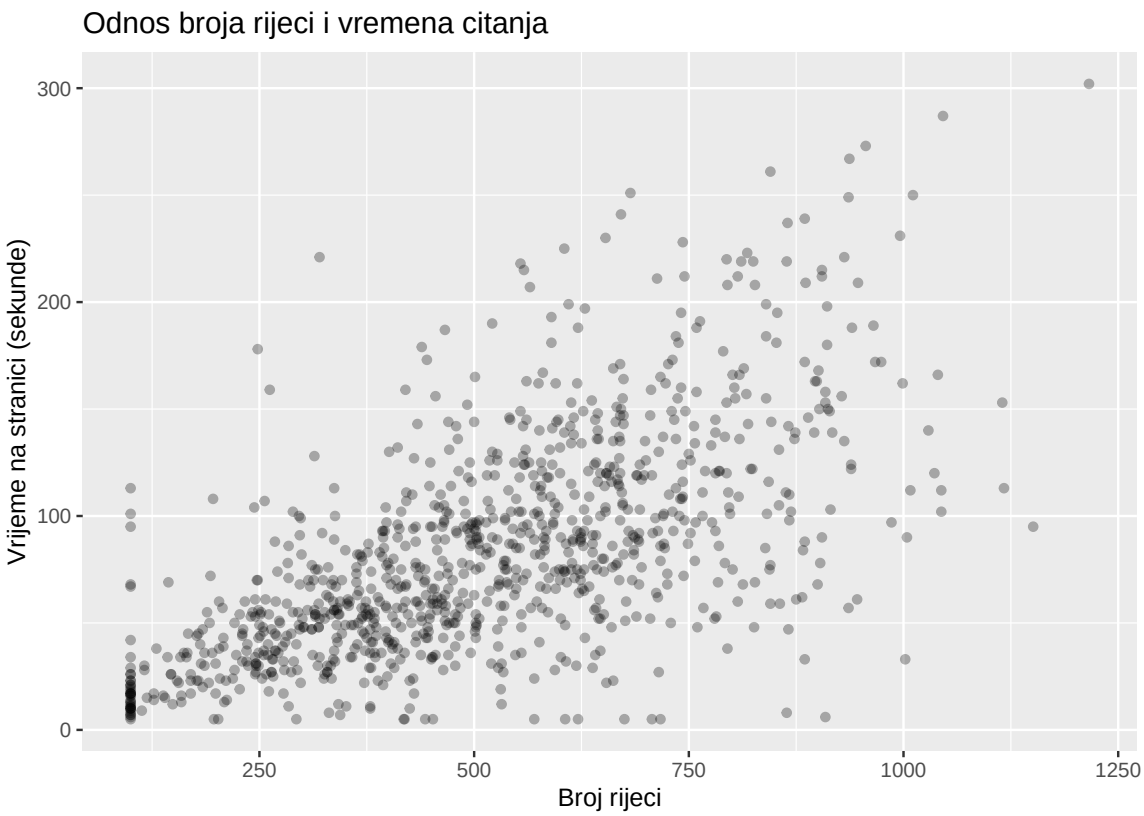
Kombinacija violin plota (za oblik distribucije) i uskog boxplota (za medijan i kvartile) daje bogat prikaz koji je ujedno i informativan i vizualno privlačan.

---

7 Točkasti grafovi (scatterplots): odnos dviju varijabli

Scatterplot je temeljni graf za vizualizaciju odnosa (korelacije) između dviju kontinuiranih varijabli. Svaka točka predstavlja jedno opažanje, s jednom varijablom na x osi i drugom na y osi.

```r
ggplot(clanci, aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.3) +
  labs(
    title = "Odnos broja riječi i vremena čitanja",
    x = "Broj riječi",
    y = "Vrijeme na stranici (sekunde)"
  )
```



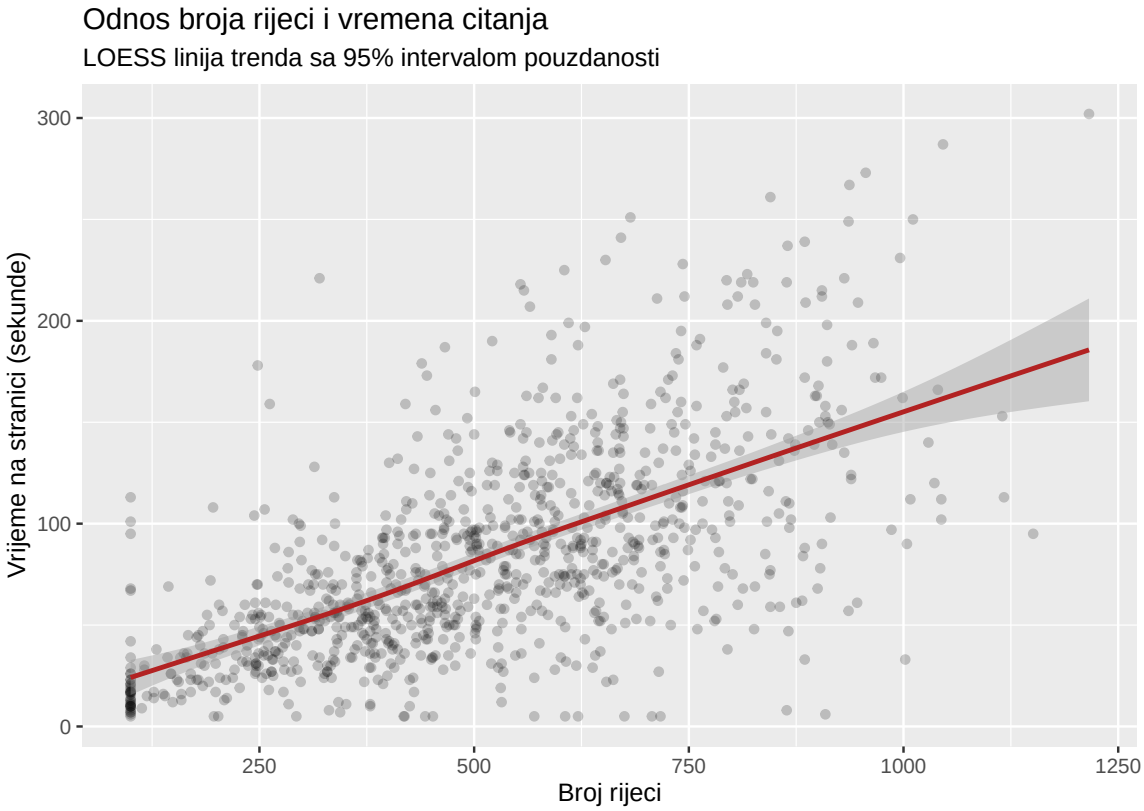
Vidimo pozitivan trend — članci s više riječi tendiraju imati duže vrijeme čitanja. Ali odnos nije savršen i postoji značajna varijabilnost. `alpha = 0.3` je bitan jer s 1000 točaka bi se bez transparentnosti mnoge preklapale i graf bi bio nečitljiv.

7.1 Dodavanje linije trenda

Funkcija `geom_smooth()` dodaje statističku liniju trenda na scatterplot. Po defaultu koristi LOESS (lokalno ponderiranu regresiju) koja je fleksibilna i prati oblik podataka.

```r
ggplot(clanci, aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.2) +
  geom_smooth(color = "firebrick", linewidth = 1) +
  labs(
```

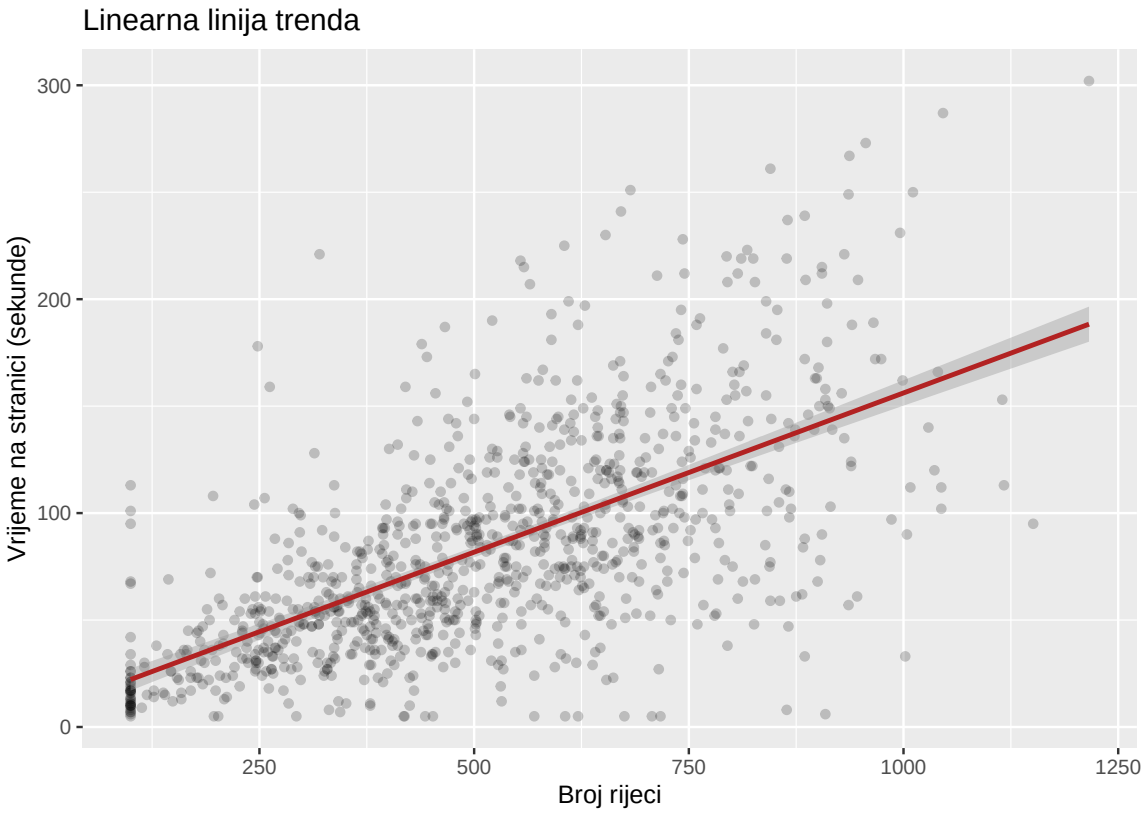
```
    title = "Odnos broja riječi i vremena čitanja",
    subtitle = "LOESS linija trenda sa 95% intervalom pouzdanosti",
    x = "Broj riječi",
    y = "Vrijeme na stranici (sekunde)"
  )
```



Sivi pojas oko linije je 95% interval pouzdanosti za procjenu trenda. Što je pojas uži, to smo sigurniji u procjenu. Na krajevima distribucije (malo i mnogo riječi) pojas je širi jer imamo manje podataka.

Za linearnu liniju trenda (onu koju smo računali pri radu na korelaciji u tjednu 4), koristite `method = "lm"`.

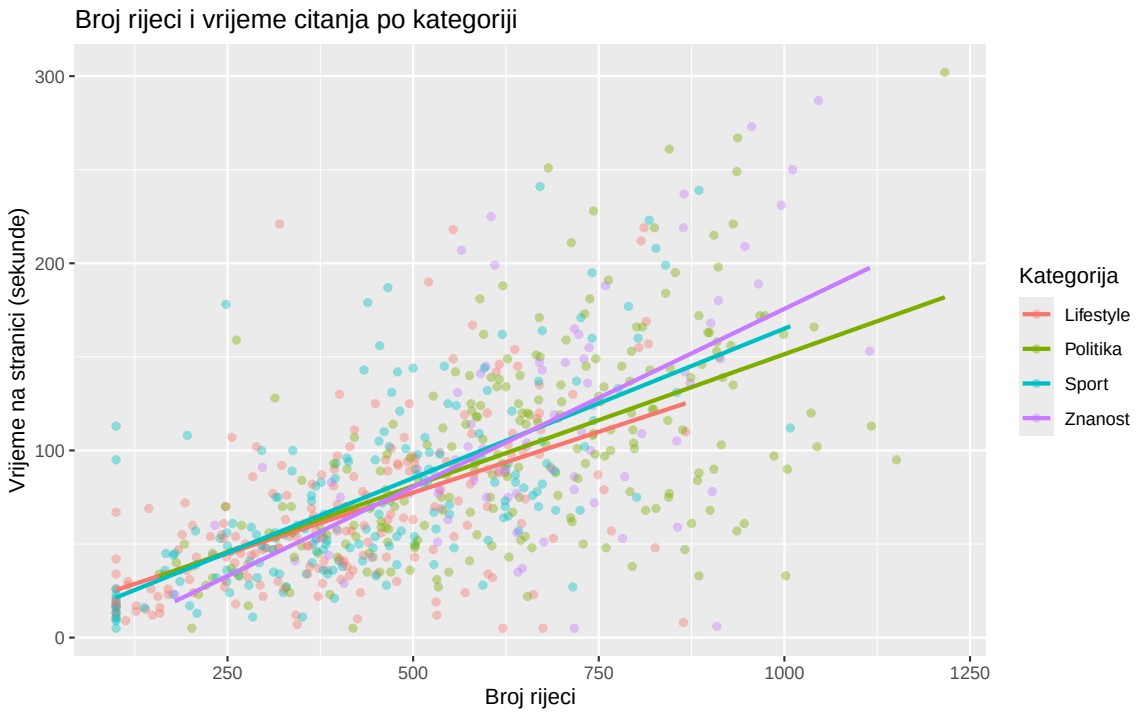
```
ggplot(clanci, aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm", color = "firebrick", linewidth = 1) +
  labs(
    title = "Linearna linija trenda",
    x = "Broj riječi",
    y = "Vrijeme na stranici (sekunde)"
  )
```



7.2 Kodiranje treće varijable bojom

Dodavanjem treće varijable kao estetike boje, scatterplot može prikazati tri dimenzije podataka na dvodimenzionalnom grafu.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost", "Lifestyle")) |>
  ggplot(aes(x = word_count, y = time_on_page, color = category)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(
    title = "Broj riječi i vrijeme čitanja po kategoriji",
    x = "Broj riječi",
    y = "Vrijeme na stranici (sekunde)",
    color = "Kategorija"
  )
```



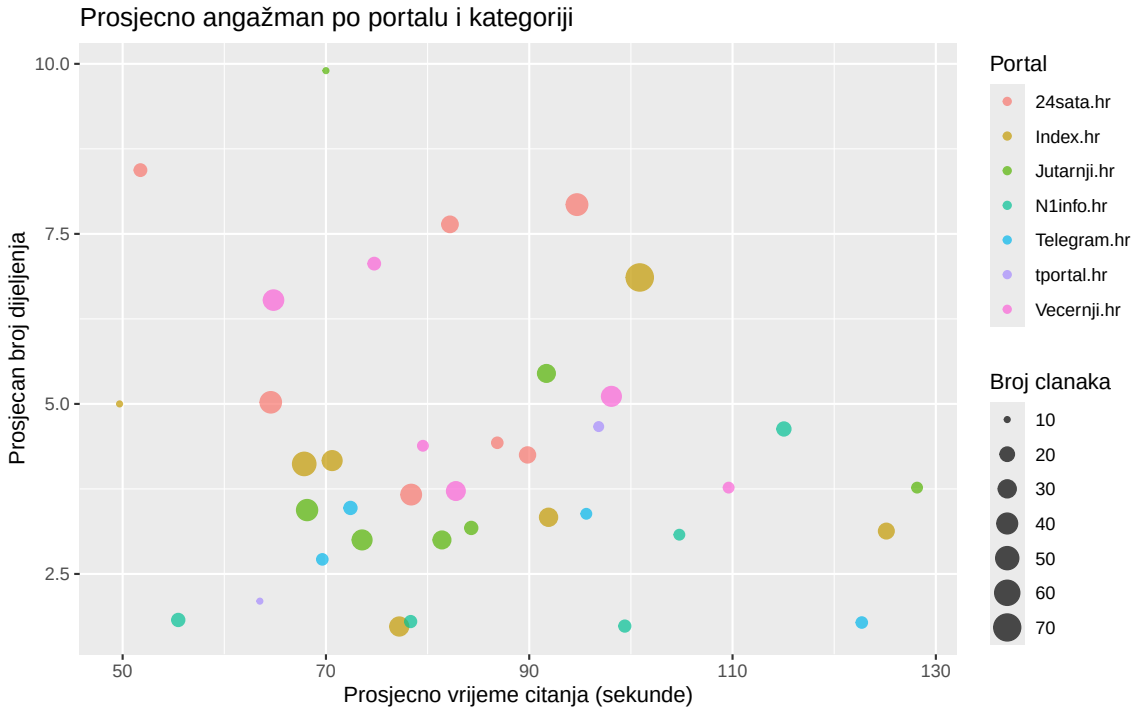
Argument `se = FALSE` uklanja interval pouzdanosti da graf ne bude pretrpan. Sada vidimo da je pozitivan odnos između broja riječi i vremena čitanja prisutan u svim kategorijama. Međutim, kategorije se razlikuju po razini (intercept) — za isti broj riječi, znanstveni članci imaju duže prosječno čitanje od lifestyle članaka.

7.3 Kodiranje veličine i oblika

Osim boje, ggplot2 nudi i druge estetike za kodiranje varijabli.

```
clanci |>
  group_by(source, category) |>
  summarise(
    prosjek_vrijeme = mean(time_on_page),
    prosjek_dijeljenja = mean(shares),
    n = n(),
    .groups = "drop"
  ) |>
  filter(n >= 10) |>
  ggplot(aes(x = prosjek_vrijeme, y = prosjek_dijeljenja,
             color = source, size = n)) +
  geom_point(alpha = 0.7) +
  labs(
    title = "Prosječno angažman po portalu i kategoriji",
    x = "Prosječno vrijeme čitanja (sekunde)",
```

```
    y = "Prosječan broj dijeljenja",
    color = "Portal",
    size = "Broj članaka"
  )
```



Ovdje smo najprije izračunali sažetke po kombinaciji portala i kategorije, a onda veličinu točke mapirali na broj članaka. Veće točke predstavljaju kombinacije s više članaka (i stoga pouzdanijim prosjekom). Ovaj tip grafa se naziva bubble chart i koristan je za prikaz tri ili četiri dimenzije podataka istovremeno.

---

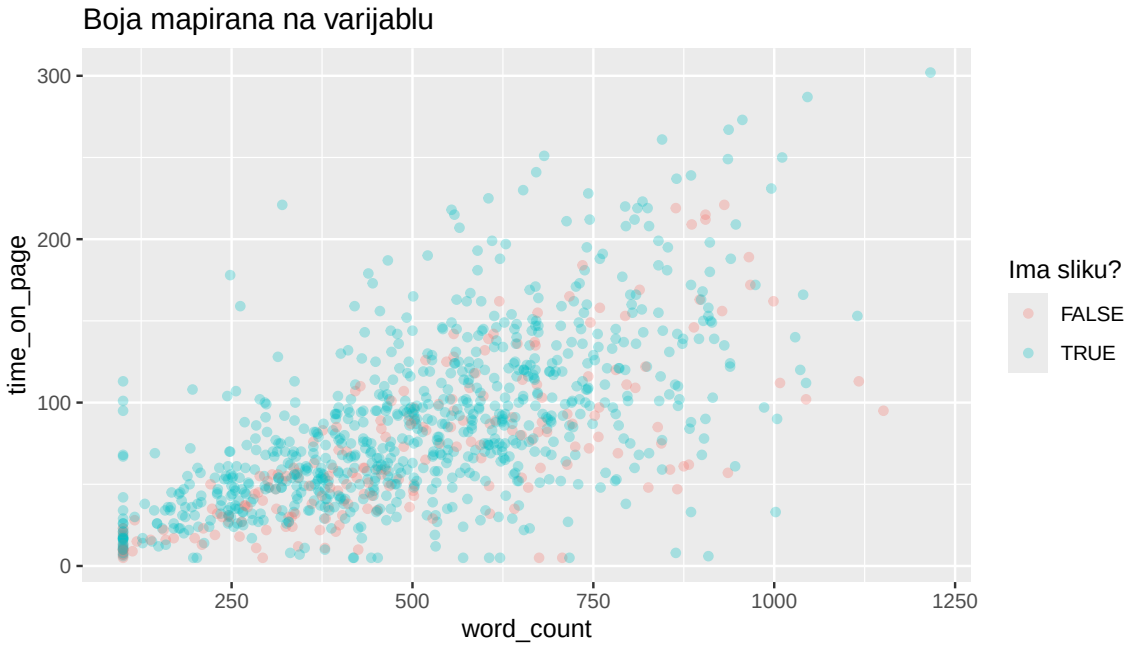
8 Estetike unutar i izvan aes()

Česta zbunjenica za početnike je razlika između estetika unutar i izvan `aes()`. Ovo je konceptualno važno razumjeti.

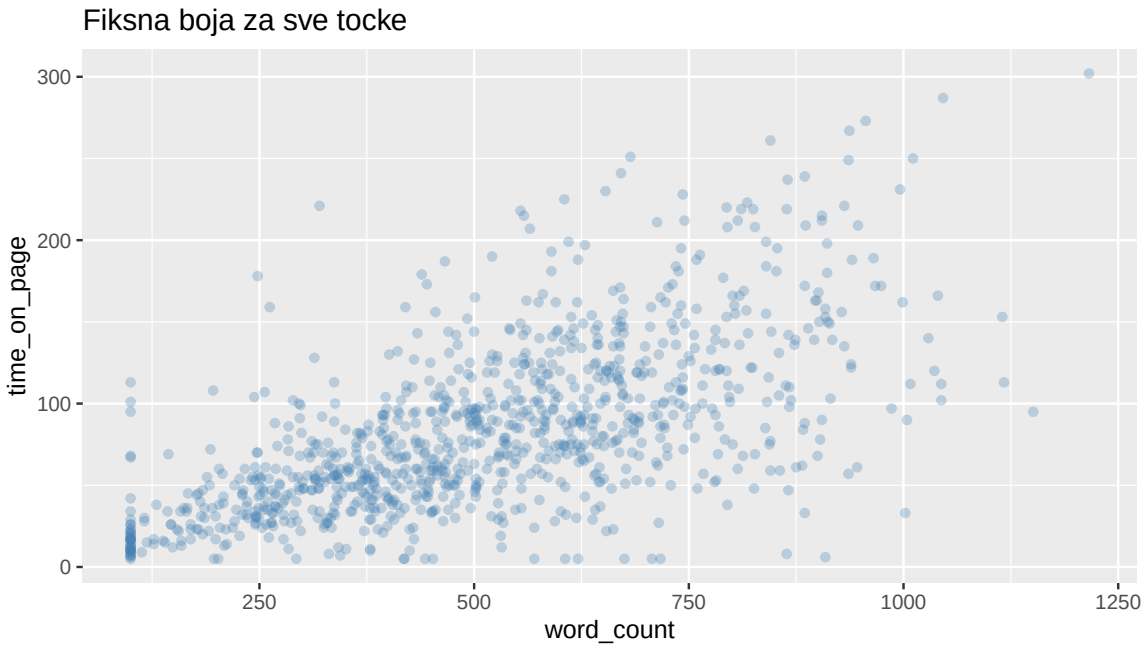
Kad stavite estetiku **unutar** `aes()`, mapirate varijablu na vizualno svojstvo. Boja ovisi o podatcima i ggplot automatski kreira legendu.

Kad stavite estetiku **izvan** `aes()`, postavljate fiksnu vrijednost za sve točke. Nema legende jer boja ne ovisi o podatcima.

```r
# UNUTAR aes(): boja ovisi o varijabli
p1 <- ggplot(clanci, aes(x = word_count, y = time_on_page, color = has_image)) +
  geom_point(alpha = 0.3) +
  labs(title = "Boja mapirana na varijablu", color = "Ima sliku?")

# IZVAN aes(): boja je fiksna za sve točke
p2 <- ggplot(clanci, aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.3, color = "steelblue") +
  labs(title = "Fiksna boja za sve točke")

p1
```



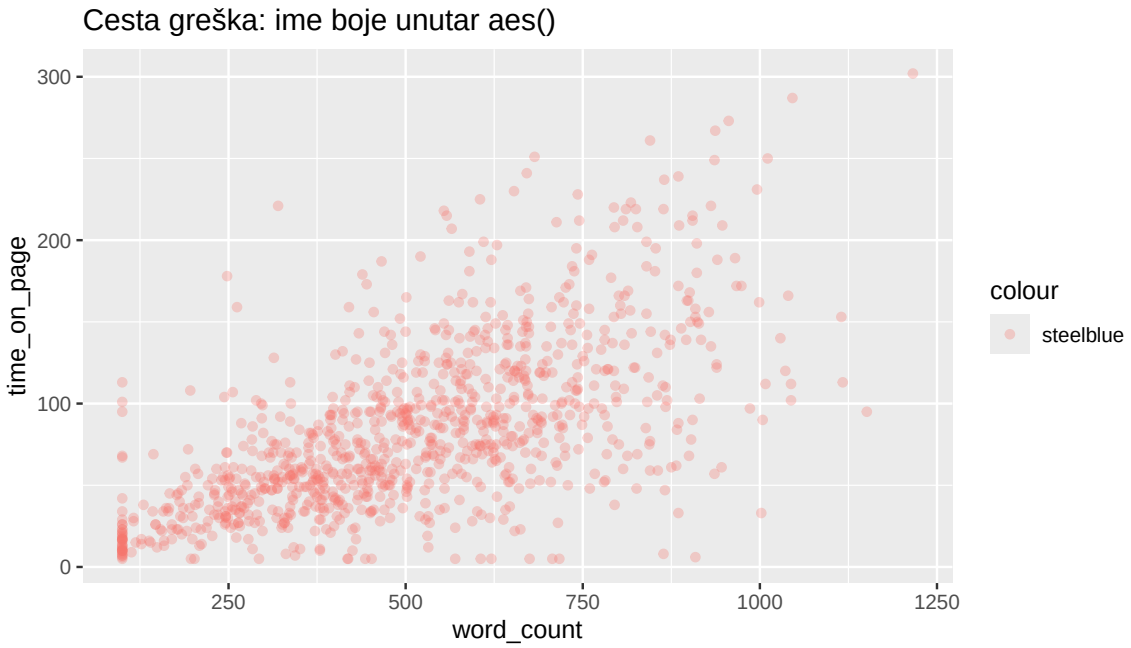

```r
p2
```



Ova razlika se proteže na sve estetike — `fill`, `color`, `size`, `shape`, `alpha`, `linewidth` — gdje varijabilne estetike idu unutar `aes()`, a fiksne vrijednosti idu izvan.

Česta greška je staviti ime boje unutar `aes()`:

```
# KRIVO: "steelblue" se tretira kao kategorija, ne kao boja
ggplot(clanci, aes(x = word_count, y = time_on_page, color = "steelblue")) +
  geom_point(alpha = 0.3) +
  labs(title = "Česta greška: ime boje unutar aes()")
```



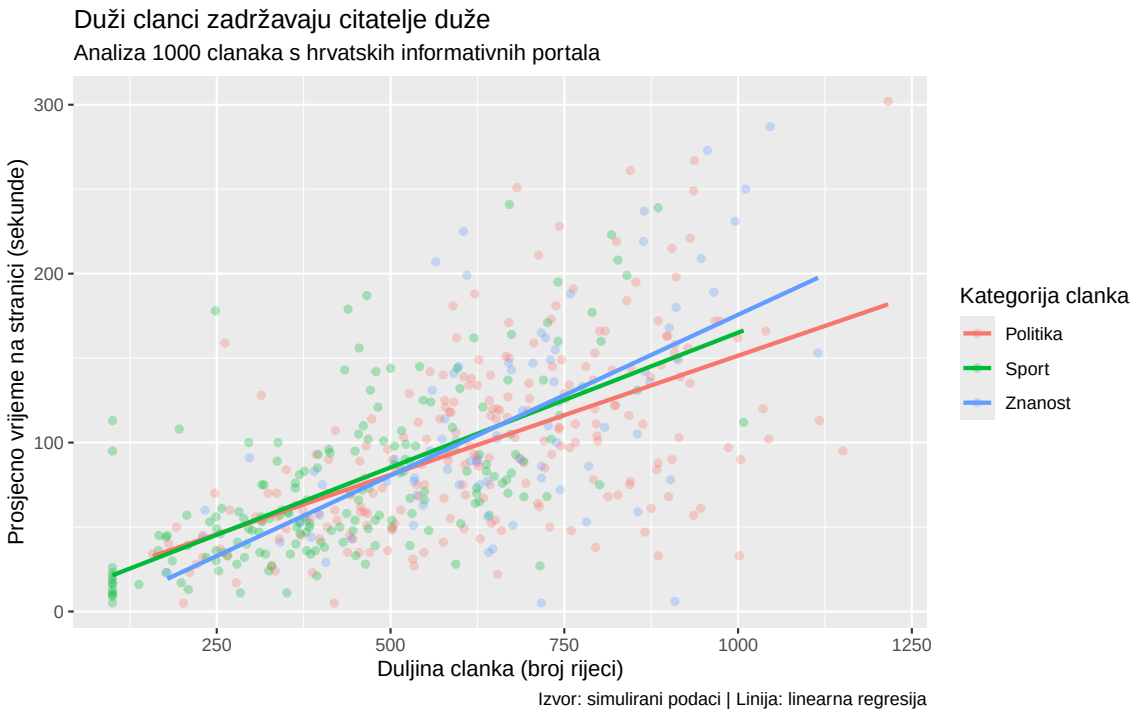
Ggplot interpretira "steelblue" kao tekstualnu varijablu s jednom kategorijom i dodjeljuje joj svoju paletu (obično crvenu) — rezultat je legenda s jednom stavkom "steelblue" obojana u boju koju ggplot sam odabere. To je jedan od najčešćih bugova kod početnaka.

---

9 labs(): naslovi, oznake i natpisi

Svaki graf koji dijete s drugima mora imati jasne oznake. Funkcija `labs()` kontrolira naslove, podnaslove, oznake osi i legende.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost")) |>
  ggplot(aes(x = word_count, y = time_on_page, color = category)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(
    title = "Duži članci zadržavaju čitatelje duže",
    subtitle = "Analiza 1000 članaka s hrvatskih informativnih portala",
    x = "Duljina članka (broj riječi)",
    y = "Prosječno vrijeme na stranici (sekunde)",
    color = "Kategorija članka",
    caption = "Izvor: simulirani podaci | Linija: linearna regresija"
  )
```



Primijetite da je naslov formuliran kao nalaz (“Duži članci zadržavaju čitatelje duže”), ne kao opis (“Odnos duljine članka i vremena čitanja”). Ovo je najbolja praksa za vizualizaciju u novinarstvu i izvještajima jer čitatelju odmah komunicira ključnu poruku. Za akademske radove, opisni naslovi su prihvatljiviji.

Argument `caption` dodaje tekst u donji desni kut grafa — koristite ga za izvor podataka ili metodološke napomene.

---

10 Brzi pregled: koji graf za koji podatak?

Do sada smo naučili četiri tipa grafova. Evo sažetka kada koristiti koji.

Histogram / density plot prikazuje distribuciju jedne kontinuirane varijable — koristite ga kad želite vidjeti oblik distribucije, identificirati outliere, provjeriti normalnost ili usporediti distribucije između grupa (npr. distribucija vremena čitanja članaka).

Stupčasti graf (bar chart) prikazuje frekvencije ili sažetke kategoričkih varijabli — koristite `geom_bar()` za automatsko prebrojavanje i `geom_col()` za prethodno izračunate sažetke (npr. broj članaka po kategoriji ili portalu).

Boxplot / violin plot uspoređuje distribucije jedne kontinuirane varijable između grupa — posebno je koristan za identifikaciju razlika u medijanima, varijabilnosti i outlierima (npr. usporedba vremena čitanja između kategorija članaka).

Scatterplot prikazuje odnos (korelaciju) između dviju kontinuiranih varijabli — dodajte `geom_smooth()` za liniju trenda i koristite boju/veličinu za kodiranje dodatnih varijabli (npr. odnos broja riječi i vremena čitanja).

```
# Praktična provjera: koje varijable imamo i što s njima prikazati?
tribble(
  ~varijable, ~tip_grafa, ~geom,
  "1 kontinuirana", "Histogram / density", "geom_histogram() / geom_density()",
  "1 kategorička", "Bar chart", "geom_bar()",
  "1 kontinuirana + 1 kategorička", "Boxplot / violin", "geom_boxplot() / geom_violin()",
  "2 kontinuirane", "Scatterplot", "geom_point() + geom_smooth()",
  "2 kategoričke", "Grupirani bar chart", "geom_bar(position = 'dodge'/'fill')"
)
```

```
# A tibble: 5 x 3
  varijable                     tip_grafa           geom
  <chr>                         <chr>               <chr>
1 1 kontinuirana                Histogram / density geom_histogram() / geom_de~
2 1 kategorička                 Bar chart           geom_bar()
3 1 kontinuirana + 1 kategorička Boxplot / violin   geom_boxplot() / geom_viol~
4 2 kontinuirane                Scatterplot         geom_point() + geom_smooth~
5 2 kategoričke                 Grupirani bar chart geom_bar(position = 'dodge~
```

Ova tablica je vaš vodič za odabir grafa. Prije nego nacrtate bilo što, postavite si pitanje — koje varijable imam i kakve su (kontinuirane ili kategoričke)? Odgovor vas automatski vodi do pravog tipa grafa.

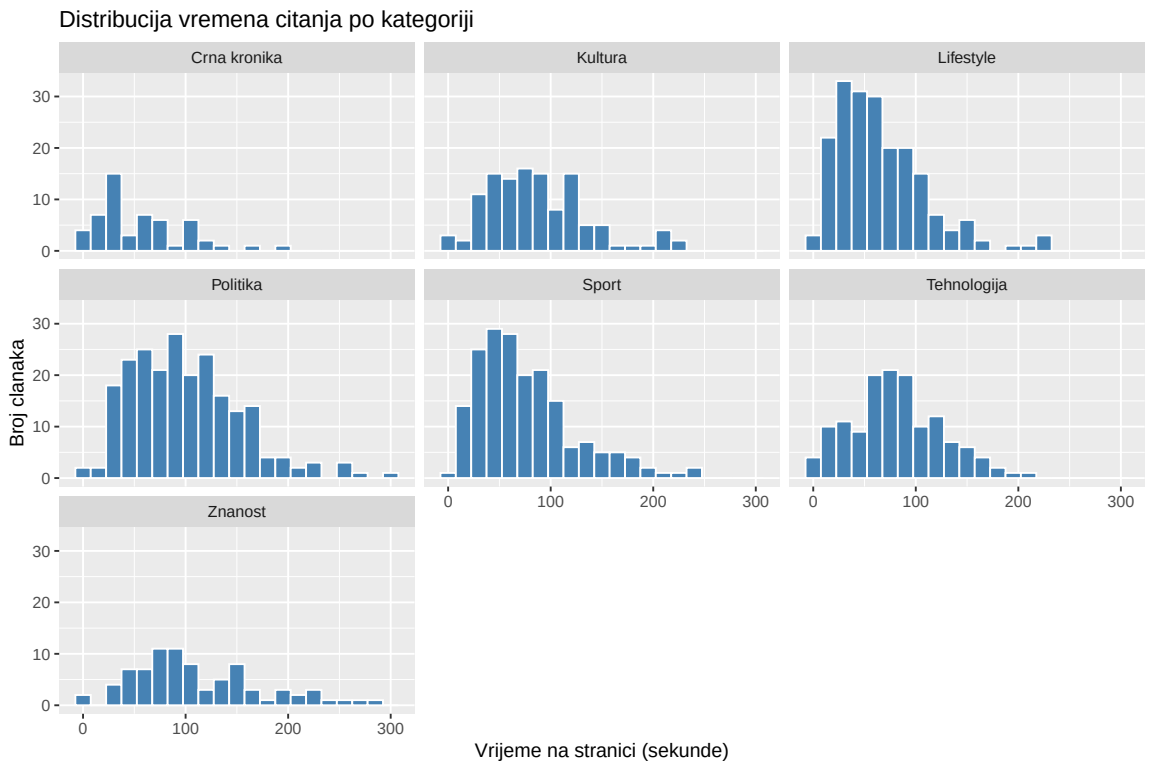
> **Podsjetnik**
>
> U prvom dijelu naučili smo četiri temeljna tipa grafova — histogram/density za distribucije, bar chart za kategorije, boxplot/violin za usporedbu grupa i scatterplot za odnos dviju varijabli. U ovom dijelu učimo kako grafove učiniti profesionalnima i prezentabilnima.

11 Facetiranje: mali višestruki grafovi

Facetiranje je jedna od najmoćnijih značajki ggplot2. Umjesto da sve grupe trpate u jedan graf s više boja, facetiranje dijeli graf na zasebne panele, po jedan za svaku grupu. Rezultat je čitljiviji jer svaki panel ima vlastite osi i nije zatrpan preklapajućim elementima.

11.1 facet_wrap(): paneli u jednom retku ili mreži

```
ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram(binwidth = 15, fill = "steelblue", color = "white") +
  facet_wrap(~category) +
  labs(
    title = "Distribucija vremena čitanja po kategoriji",
    x = "Vrijeme na stranici (sekunde)",
    y = "Broj članaka"
  )
```

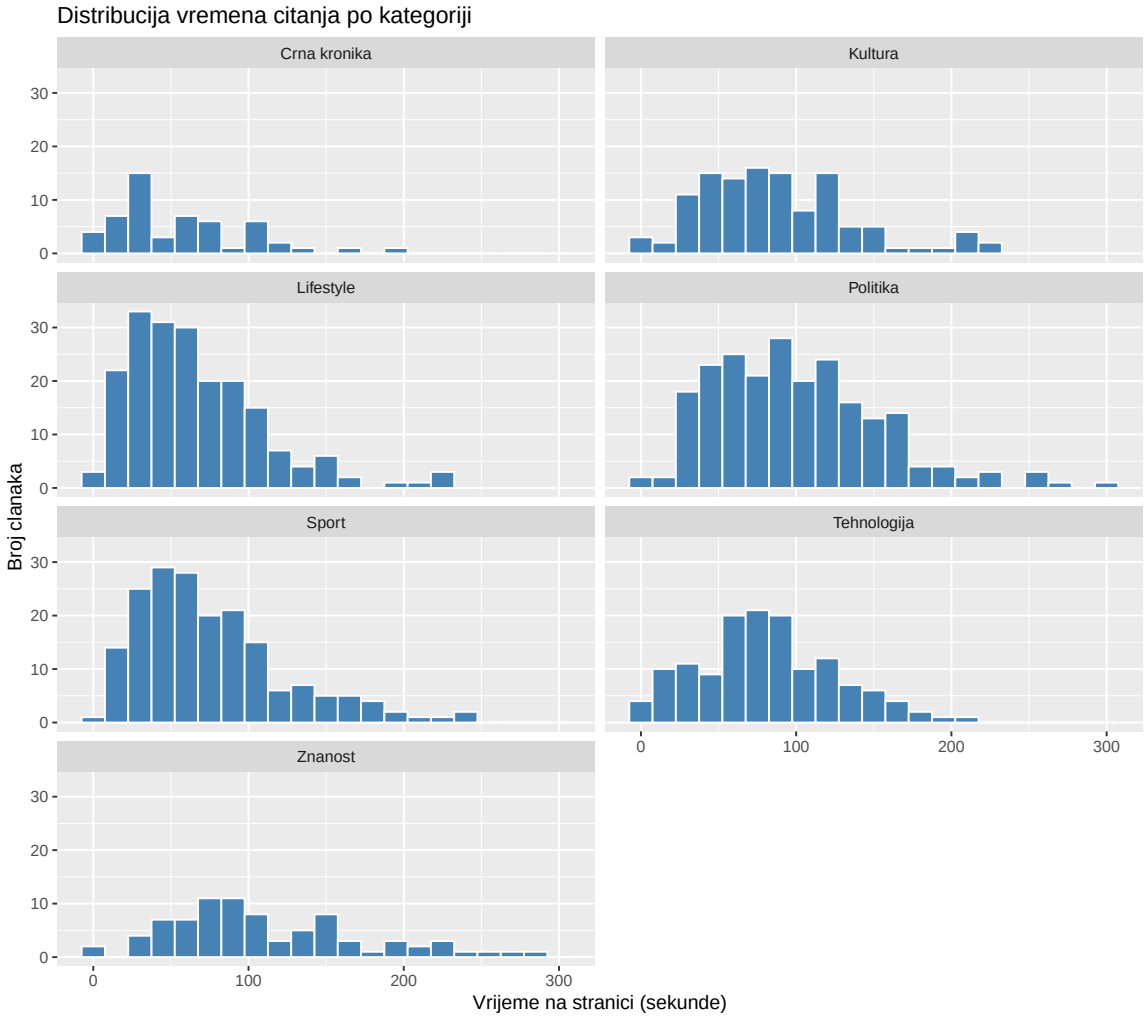


Sintaksa `facet_wrap(~category)` govori ggplotu da napravi zaseban panel za svaku razinu varijable `category`. Tilda (`~`) je obavezna i čita se kao "po", što znači podijeli po kategoriji. Paneli se automatski slažu u mrežu.

Argument `ncol` kontrolira broj stupaca u mreži.

```
ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram(binwidth = 15, fill = "steelblue", color = "white") +
  facet_wrap(~category, ncol = 2) +
  labs(
    title = "Distribucija vremena čitanja po kategoriji",
    x = "Vrijeme na stranici (sekunde)",
```

```
    y = "Broj članaka"
  )
```



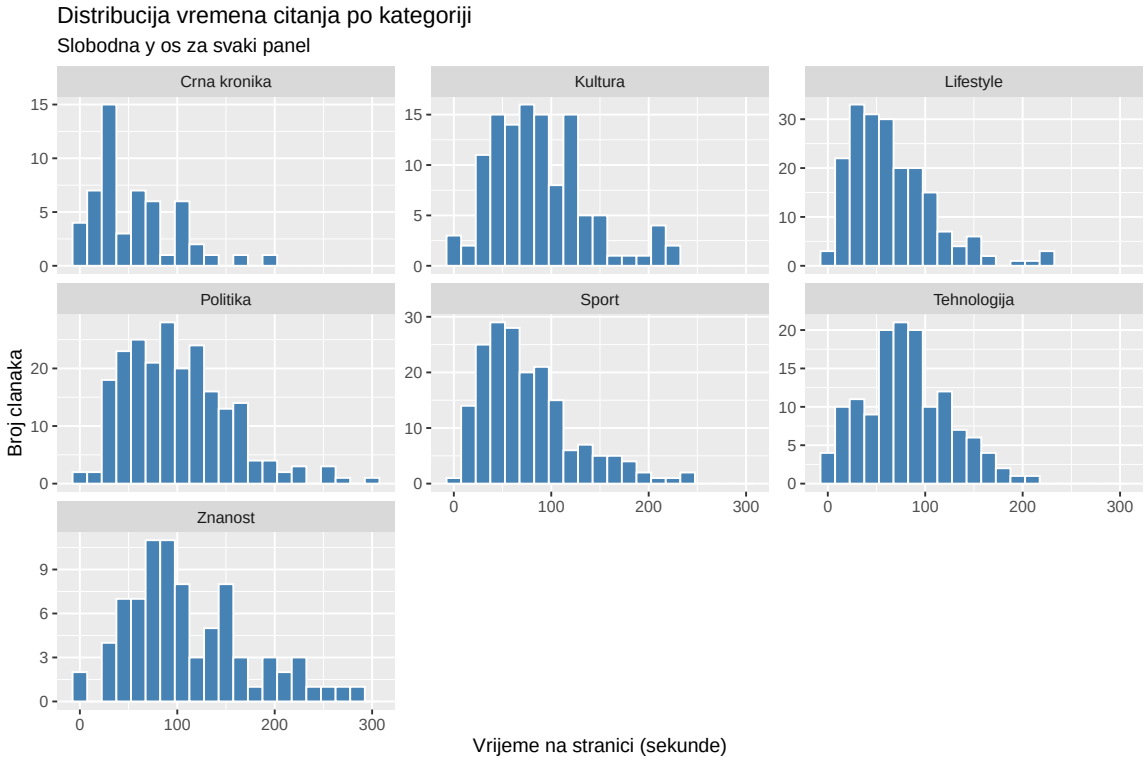
S `ncol = 2` dobivamo dva stupca panela, što je čitljivije kad imate mnogo kategorija jer su paneli širi i histogram je jasniji.

11.2 Slobodne osi u facetima

Po defaultu, svi paneli dijele iste osi. Ovo je dobro za usporedbu apsolutnih vrijednosti, ali ponekad želite da svaki panel ima vlastitu skalu (na primjer, kad se grupe drastično razlikuju po broju opažanja).

```
ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram(binwidth = 15, fill = "steelblue", color = "white") +
  facet_wrap(~category, scales = "free_y") +
```

```
  labs(
    title = "Distribucija vremena čitanja po kategoriji",
    subtitle = "Slobodna y os za svaki panel",
    x = "Vrijeme na stranici (sekunde)",
    y = "Broj članaka"
  )
```



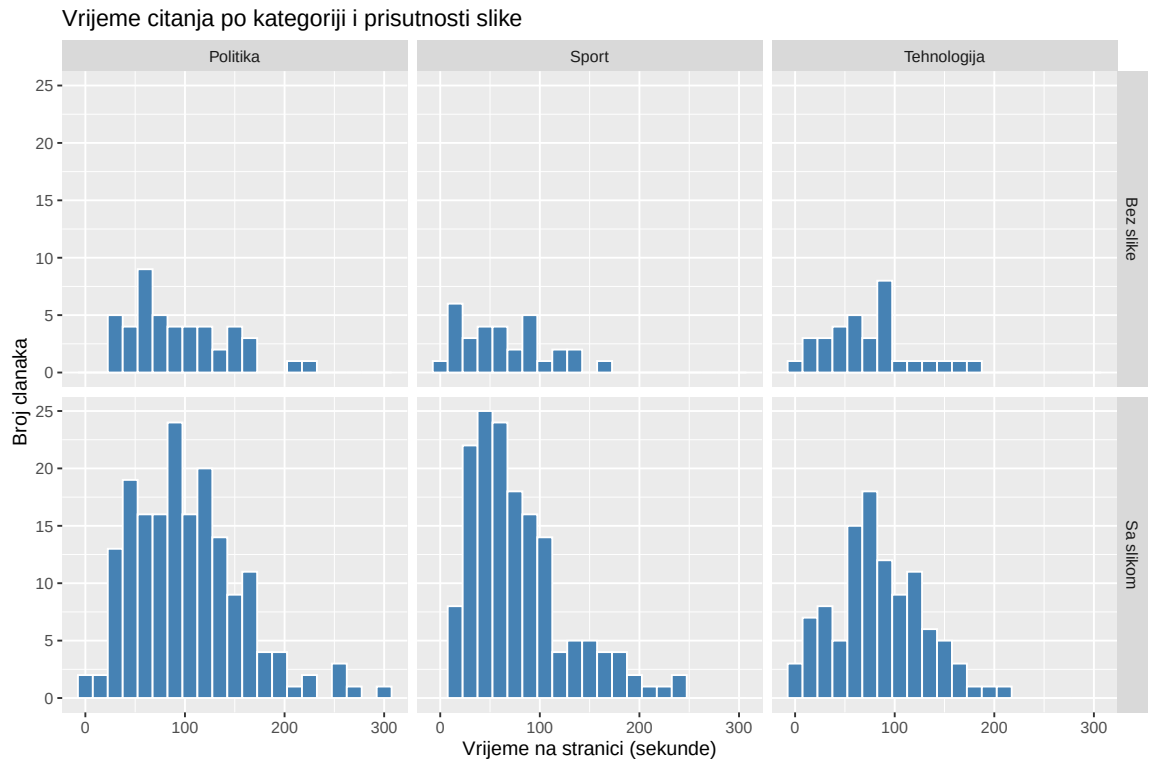
Opcije za `scales` uključuju `"fixed"` (default, iste osi), `"free_x"` (slobodna x os), `"free_y"` (slobodna y os) i `"free"` (obje slobodne). Koristite slobodne osi samo kad imate dobar razlog jer otežavaju izravnu usporedbu između panela.

11.3 facet_grid(): paneli u matrici dviju varijabli

Dok `facet_wrap()` dijeli po jednoj varijabli i slaže panele u mrežu, `facet_grid()` kreira matricu panela po dvjema varijablama — jedna definira retke, druga stupce.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Tehnologija")) |>
  mutate(ima_sliku = if_else(has_image, "Sa slikom", "Bez slike")) |>
  ggplot(aes(x = time_on_page)) +
  geom_histogram(binwidth = 15, fill = "steelblue", color = "white") +
  facet_grid(ima_sliku ~ category) +
```

```
  labs(
    title = "Vrijeme čitanja po kategoriji i prisutnosti slike",
    x = "Vrijeme na stranici (sekunde)",
    y = "Broj članaka"
  )
```



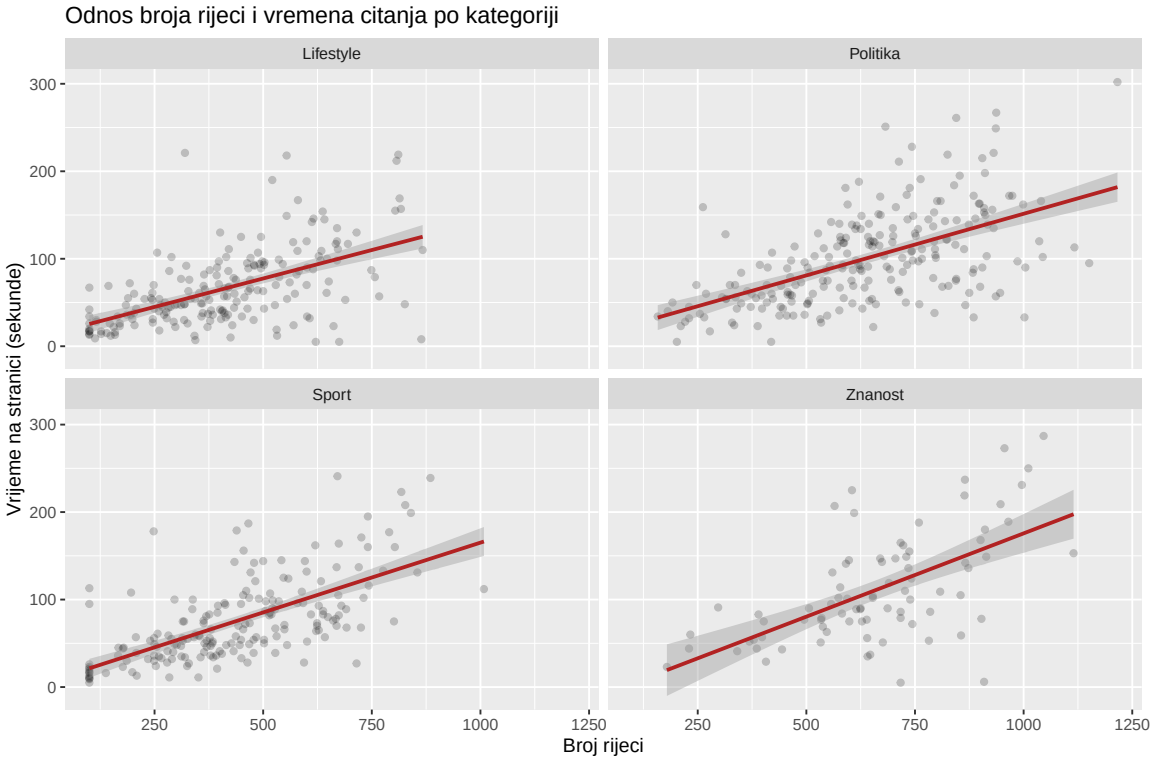
Sintaksa `facet_grid(retci ~ stupci)` postavlja varijable u redove i stupce matrice. Ovo je idealno za prikaz interakcije dviju kategoričkih varijabli jer možete uspoređivati kako vertikalno (unutar iste kategorije, sa i bez slike) tako i horizontalno (između kategorija, za isti format).

11.4 Facetiranje scatterplota

Facetiranje radi sa svakim tipom grafa, ne samo s histogramima.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost", "Lifestyle")) |>
  ggplot(aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm", color = "firebrick") +
  facet_wrap(~category) +
  labs(
```

```
    title = "Odnos broja riječi i vremena čitanja po kategoriji",
    x = "Broj riječi",
    y = "Vrijeme na stranici (sekunde)"
  )
```



Svaki panel ima vlastitu regresijsku liniju, pa možemo vidjeti je li odnos sličan u svim kategorijama ili se razlikuje. Ovo je vizualna prethodnica interakcije u regresijskoj analizi (tjedan 14).

> **Praktični savjet**
>
> Facetiranje je gotovo uvijek bolje od preklapanja mnogo grupa u jednom grafu. Kad imate više od tri ili četiri grupe, graf s jednim panelom postaje nečitljiv bez obzira koliko pažljivo birate boje i transparentnost. Facet_wrap s 6 ili 8 panela je čitljiviji od jednog pretrpanog grafa.

---

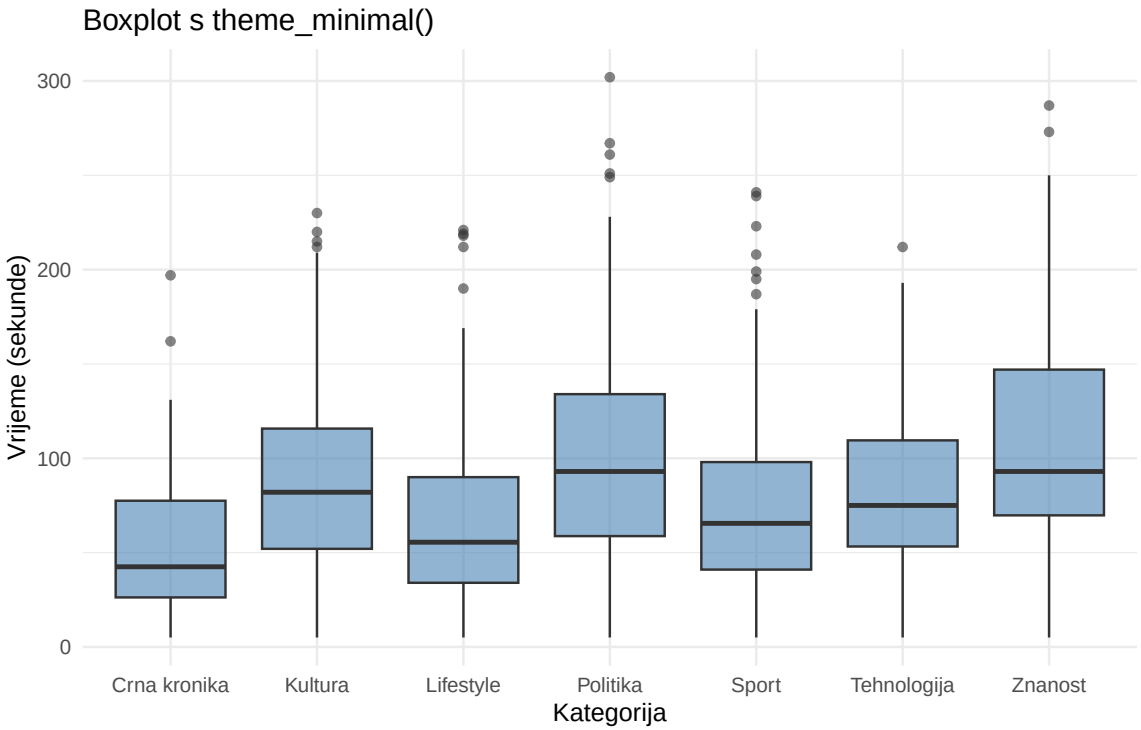
12 Teme: vizualni izgled grafa

Svaki ggplot2 graf ima temu koja kontrolira sve vizualne elemente koji nisu podaci — pozadinu, mrežu (grid lines), fontove, margine, poziciju legende i slično. Defaultna tema

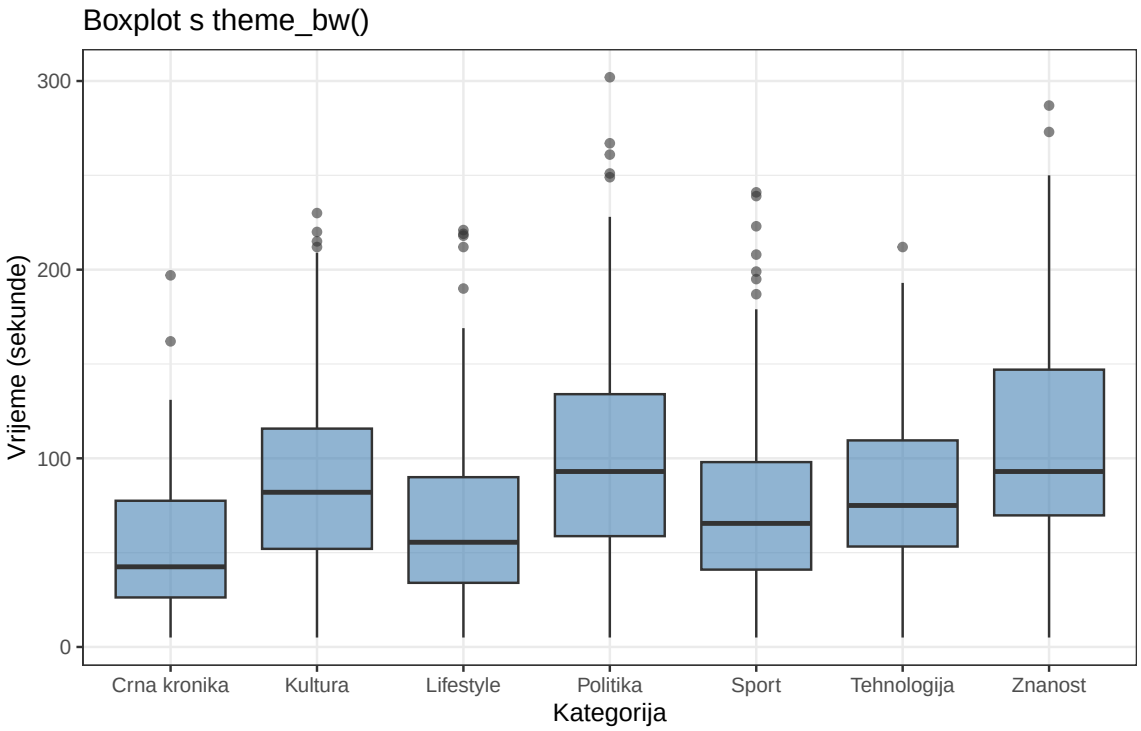
(`theme_gray()`) ima sivu pozadinu s bijelom mrežom. Za profesionalni rad, gotovo uvijek ćete koristiti neku drugu temu.

12.1 Ugrađene teme

```
ggplot(clanci, aes(x = category, y = time_on_page)) +
  geom_boxplot(fill = "steelblue", alpha = 0.6) +
  theme_minimal() +
  labs(
    title = "Boxplot s theme_minimal()",
    x = "Kategorija",
    y = "Vrijeme (sekunde)"
  )
```




```
ggplot(clanci, aes(x = category, y = time_on_page)) +
  geom_boxplot(fill = "steelblue", alpha = 0.6) +
  theme_bw() +
  labs(
    title = "Boxplot s theme_bw()",
    x = "Kategorija",
    y = "Vrijeme (sekunde)"
  )
```



`theme_minimal()` je čista tema bez okvira i s minimalnom mrežom. Odlična za prezentacije i izvještaje. `theme_bw()` je slična ali s crnim okvirom oko grafa. Obje su popularnije od defaultne sive teme.

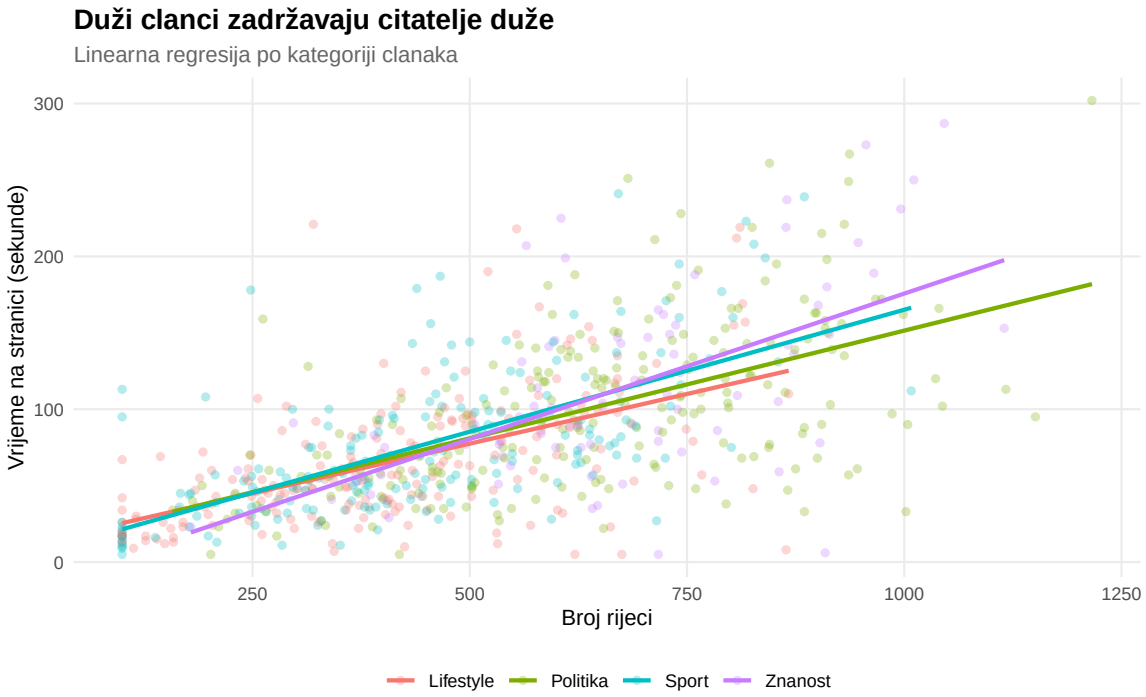
Ostale ugrađene teme uključuju `theme_classic()` (samo osi, bez mreže, tradicionalan izgled), `theme_light()` (svijetla pozadina s tankom mrežom) i `theme_void()` (prazan prostor, korisno za karte i dijagrame).

12.2 Prilagodba s theme()

Funkcija `theme()` omogućuje prilagodbu pojedinačnih vizualnih elemenata. Ovo je detaljni alat za fino podešavanje izgleda.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost", "Lifestyle")) |>
  ggplot(aes(x = word_count, y = time_on_page, color = category)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = FALSE) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    plot.subtitle = element_text(size = 11, color = "grey40"),
    axis.title = element_text(size = 11),
    legend.position = "bottom",
```

```
    panel.grid.minor = element_blank()
  ) +
  labs(
    title = "Duži članci zadržavaju čitatelje duže",
    subtitle = "Linearna regresija po kategoriji članaka",
    x = "Broj riječi",
    y = "Vrijeme na stranici (sekunde)",
    color = NULL
  )
```



Raščlanimo `theme()` argumente — `element_text()` kontrolira fontove (veličinu, bold/italic, boju), dok `element_blank()` potpuno uklanja element (u ovom slučaju minor grid linije). `legend.position = "bottom"` premješta legendu ispod grafa. Postavljanje `color = NULL` u `labs()` uklanja naslov legende kad je očit iz konteksta.

Redoslijed je bitan: prvo dodajte ugrađenu temu (`theme_minimal()`), pa onda vlastite prilagodbe s `theme()`. Obrnuti redoslijed ne bi radio jer bi ugrađena tema pregazila vaše prilagodbe.

12.3 Postavljanje globalne teme

Ako želite da svi grafovi u dokumentu koriste istu temu, postavite je globalno na početku.

```
# Postavljanje globalne teme za sve grafove
theme_set(
  theme_minimal() +
    theme(
      plot.title = element_text(size = 13, face = "bold"),
      plot.subtitle = element_text(size = 10, color = "grey40"),
      panel.grid.minor = element_blank()
    )
)
```

Od ovog trenutka, svaki graf u dokumentu automatski koristi ovu temu. Ne morate je dodavati svakom grafu posebno. Ovo osigurava vizualnu konzistentnost kroz cijeli izvještaj ili prezentaciju.

---

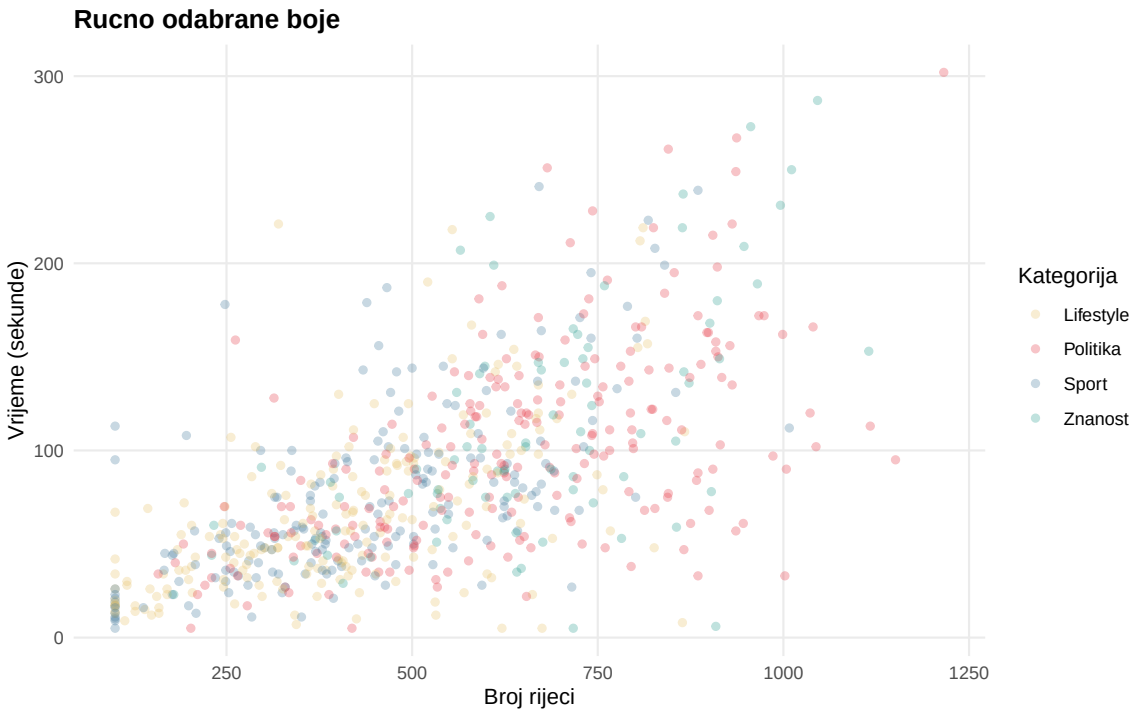
13 Skale boja

Defaultne boje u ggplot2 su funkcionalne ali ne uvijek idealne. Paket nudi više sustava boja prilagođenih različitim potrebama.

13.1 Ručni odabir boja

Za kategoričke varijable s nekoliko razina, ponekad je najbolje ručno odrediti boje.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost", "Lifestyle")) |>
  ggplot(aes(x = word_count, y = time_on_page, color = category)) +
  geom_point(alpha = 0.3) +
  scale_color_manual(values = c(
    "Politika" = "#e63946",
    "Sport" = "#457b9d",
    "Znanost" = "#2a9d8f",
    "Lifestyle" = "#e9c46a"
  )) +
  labs(
    title = "Ručno odabrane boje",
    x = "Broj riječi",
    y = "Vrijeme (sekunde)",
    color = "Kategorija"
  )
```

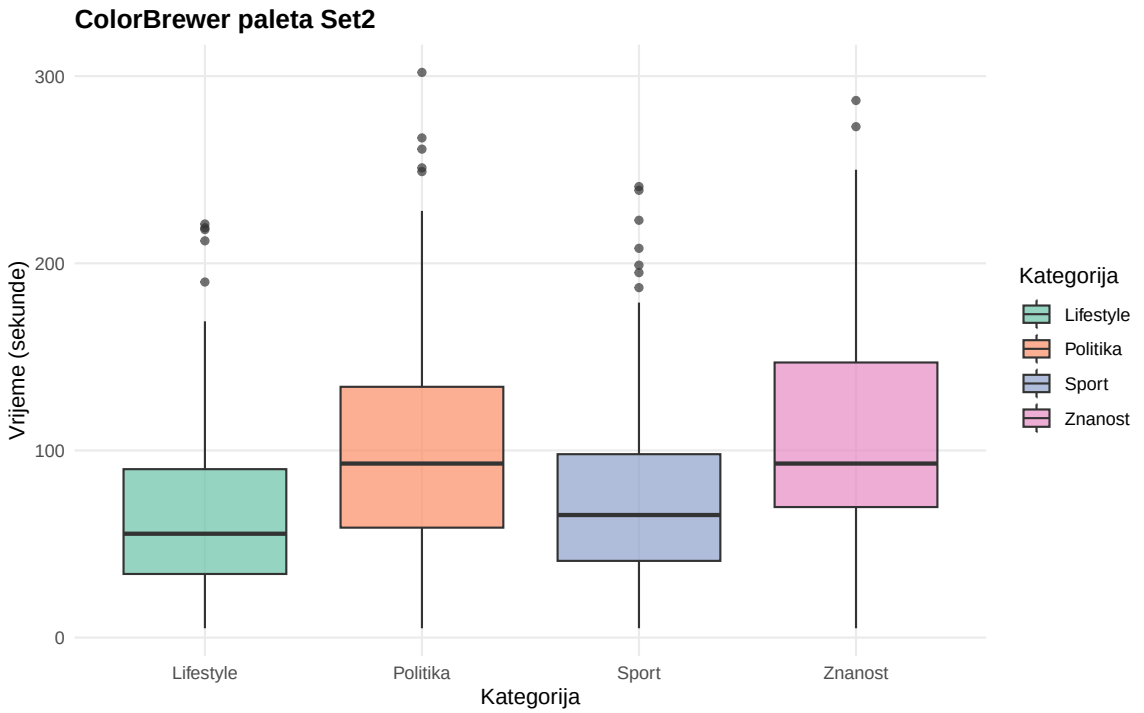


`scale_color_manual()` (za boju linija i točaka) i `scale_fill_manual()` (za boju ispune) primaju imenovani vektor boja. Prednost ručnog odabira je potpuna kontrola, ali zahtijeva poznavanje hex kodova boja ili korištenje alata poput coolors.co za odabir usklađenih paleta.

13.2 ColorBrewer palete

Paket RColorBrewer nudi provjerene palete dizajnirane za kartografiju i vizualizaciju podataka. Dostupne su kroz `scale_color_brewer()` i `scale_fill_brewer()`.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost", "Lifestyle")) |>
  ggplot(aes(x = category, y = time_on_page, fill = category)) +
  geom_boxplot(alpha = 0.7) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "ColorBrewer paleta Set2",
    x = "Kategorija",
    y = "Vrijeme (sekunde)",
    fill = "Kategorija"
  )
```

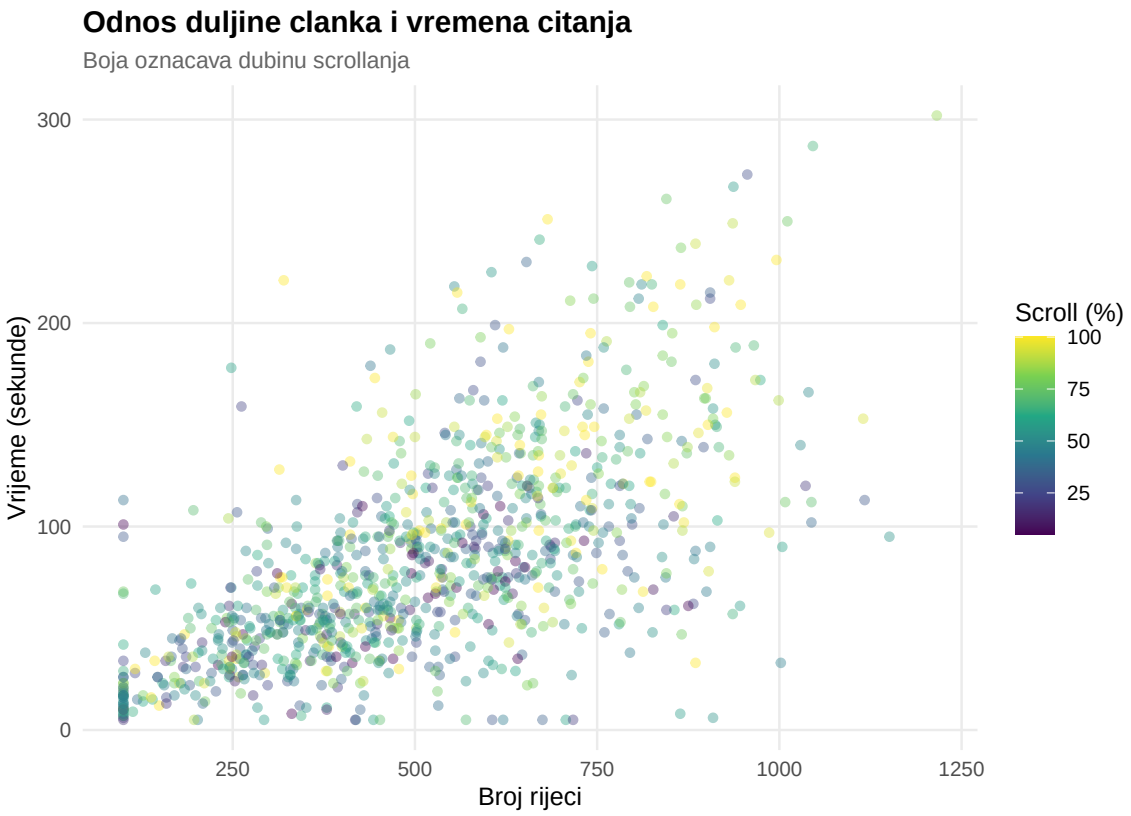


Brewer palete dolaze u tri tipa — **kvalitativne** za kategorije (npr. “Set1”, “Set2”, “Dark2”, “Pastel1”), **sekvencijalne** za gradijent od niske do visoke vrijednosti (npr. “Blues”, “Reds”, “YlOrRd”) i **divergentne** za vrijednosti koje se razilaze od sredine (npr. “RdBu”, “PRGn”).

13.3 Viridis palete

Viridis palete su dizajnirane da budu perceptualno uniformne (jednaki koraci u boji odgovaraju jednakim koracima u podatcima), čitljive u crno-bijelom ispisu i pristupačne osobama s poremećajem vida boja.

```
ggplot(clanci, aes(x = word_count, y = time_on_page, color = scroll_depth)) +
  geom_point(alpha = 0.4) +
  scale_color_viridis_c() +
  labs(
    title = "Odnos duljine članka i vremena čitanja",
    subtitle = "Boja označava dubinu scrollanja",
    x = "Broj riječi",
    y = "Vrijeme (sekunde)",
    color = "Scroll (%)"
  )
```



`scale_color_viridis_c()` koristi kontinuiranu viridis paletu za numeričke varijable, dok `scale_color_viridis_d()` koristi diskretnu verziju za kategoričke varijable. Opcija `option` bira između varijanti: "viridis" (default, plavo-zeleno-žuta), "magma" (crno-crveno-žuta), "plasma", "inferno" i "turbo".

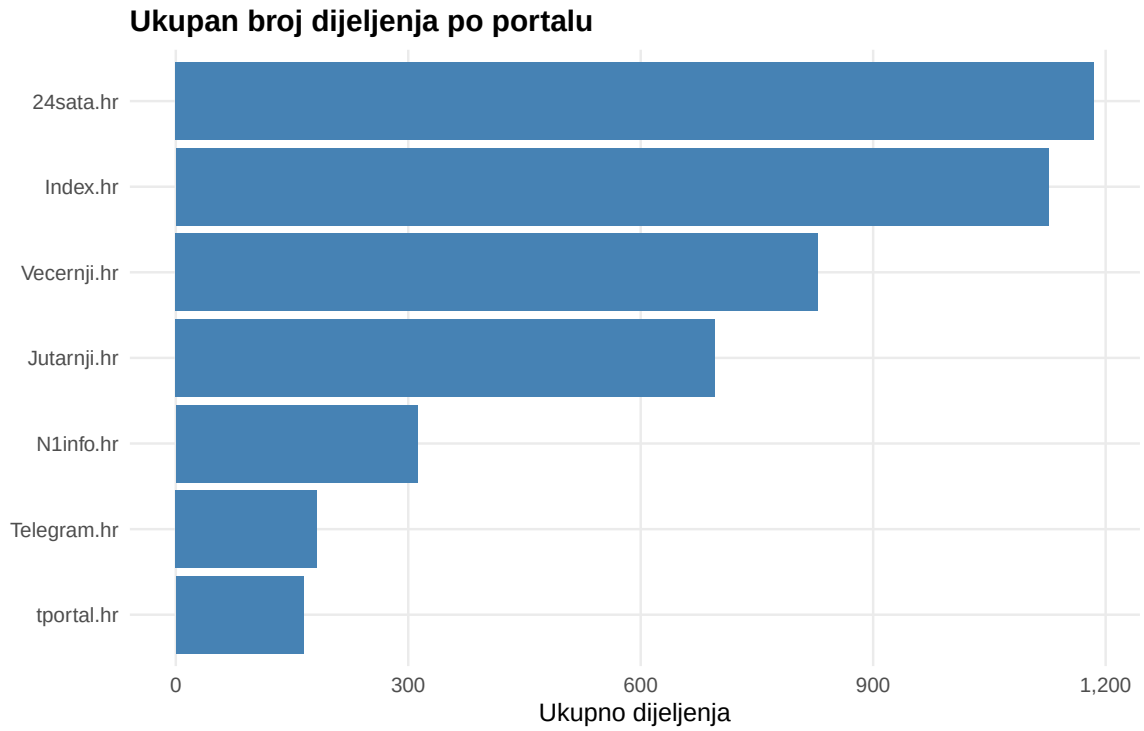
> **!** Važna napomena
>
> Boja ima dva kanala u ggplot2 — `color` za rubove i linije i `fill` za ispunu. Svaki ima vlastite scale funkcije. Za boxplot koristite `scale_fill_*()`, za scatterplot `scale_color_*()`, za bar chart `scale_fill_*()`. Ako koristite krivi kanal, boja se neće promijeniti i nećete dobiti grešku, samo prazan vizualni rezultat.

---

14 Formatiranje osi

Ponekad defaultne oznake na osima nisu optimalne. Paket scales (automatski učitan s tidyverse) pruža pomoćne funkcije za formatiranje.

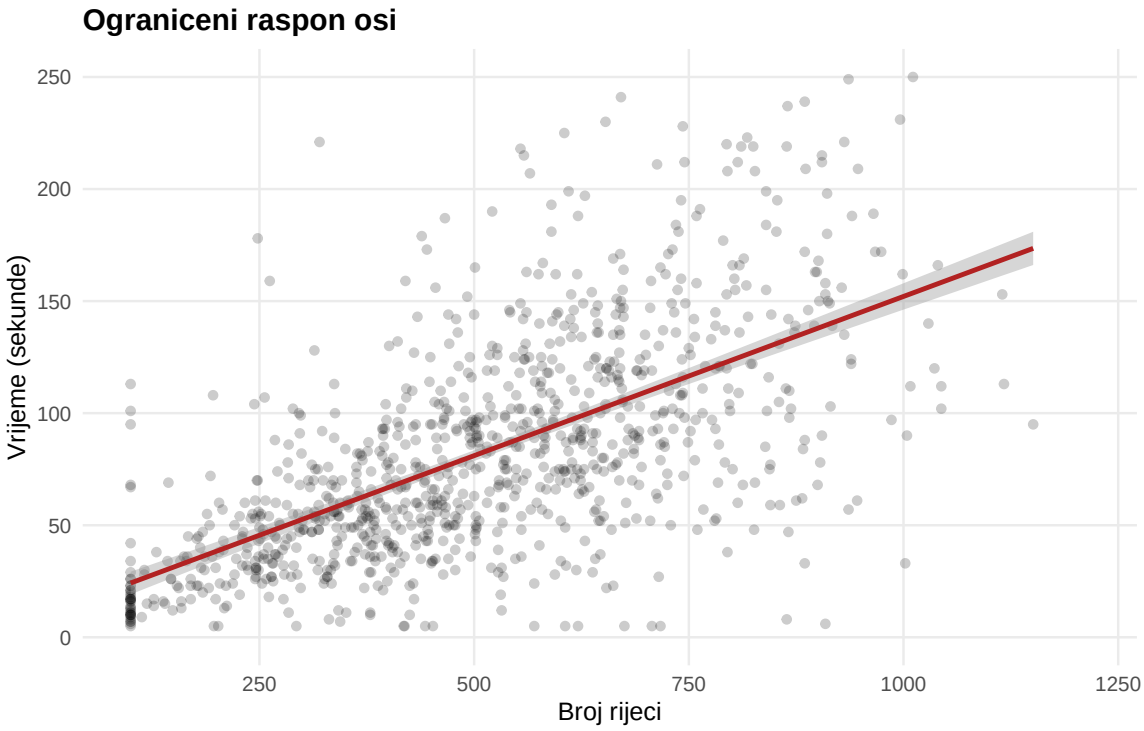
```r
library(scales)

clanci |>
  group_by(source) |>
  summarise(ukupno_dijeljenja = sum(shares), .groups = "drop") |>
  mutate(source = fct_reorder(source, ukupno_dijeljenja)) |>
  ggplot(aes(x = source, y = ukupno_dijeljenja)) +
  geom_col(fill = "steelblue") +
  scale_y_continuous(labels = label_comma()) +
  coord_flip() +
  labs(
    title = "Ukupan broj dijeljenja po portalu",
    x = NULL,
    y = "Ukupno dijeljenja"
  )
```



Funkcija `label_comma()` formatira brojeve s tisućicama (1,000 umjesto 1000). Druge korisne funkcije iz paketa scales uključuju `label_percent()` za postotke, `label_dollar()` za valute i `label_number(suffix = " min")` za dodavanje mjernih jedinica.

14.1 Kontrola raspona osi

```
ggplot(clanci, aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm", color = "firebrick") +
  scale_x_continuous(breaks = seq(0, 1500, by = 250)) +
  scale_y_continuous(limits = c(0, 250)) +
  labs(
    title = "Ograničeni raspon osi",
    x = "Broj riječi",
    y = "Vrijeme (sekunde)"
  )
```



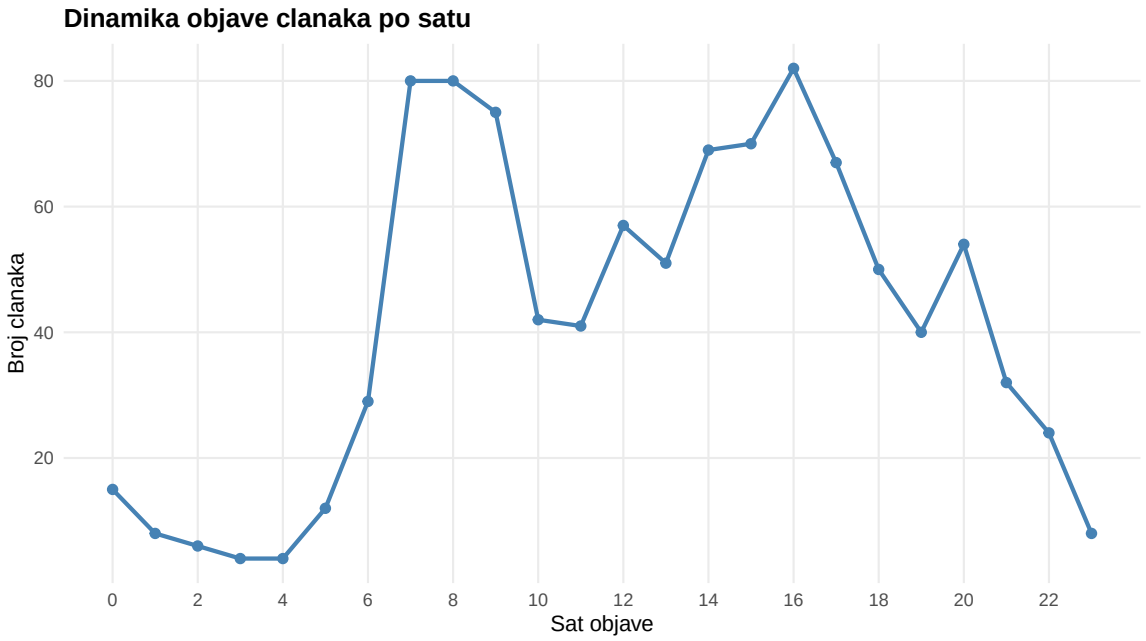
`breaks` kontrolira gdje se pojavljuju oznake na osi, dok `limits` ograničava raspon osi (točke izvan raspona se uklanjaju iz grafa). Koristite `coord_cartesian(ylim = c(0, 250))` umjesto `limits` ako želite "zumirati" bez uklanjanja podataka, jer `coord_cartesian()` samo sužava prikaz dok `limits` zaista filtrira podatke prije nego ih ggplot obradi (što može utjecati na linije trenda).

---

15 Linijski grafovi: trendovi i serije

Linijski grafovi su prirodan izbor za podatke koji imaju redoslijed, posebno vremenski. Pogledajmo distribuciju objava po satu.

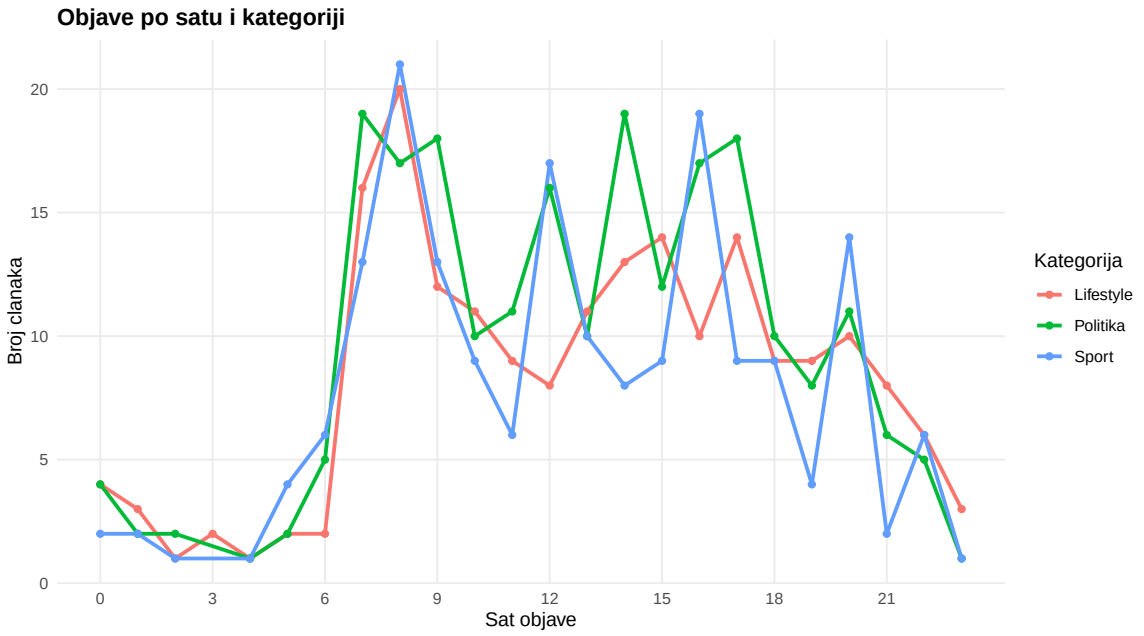
```
clanci |>
  count(publish_hour) |>
  ggplot(aes(x = publish_hour, y = n)) +
  geom_line(color = "steelblue", linewidth = 1) +
  geom_point(color = "steelblue", size = 2) +
  scale_x_continuous(breaks = seq(0, 23, by = 2)) +
  labs(
    title = "Dinamika objave članaka po satu",
    x = "Sat objave",
    y = "Broj članaka"
  )
```



Kombinacija `geom_line()` i `geom_point()` je uobičajena — linije pokazuju trend, dok točke označavaju stvarne podatke. Vidimo jasne vrhunce u jutarnjim satima i kasno popodne, što odgovara redakcijskim ciklusima.

15.1 Više linija u jednom grafu

```r
clanci |>
  filter(category %in% c("Politika", "Sport", "Lifestyle")) |>
  count(publish_hour, category) |>
  ggplot(aes(x = publish_hour, y = n, color = category)) +
  geom_line(linewidth = 1) +
  geom_point(size = 1.5) +
  scale_x_continuous(breaks = seq(0, 23, by = 3)) +
  labs(
    title = "Objave po satu i kategoriji",
    x = "Sat objave",
    y = "Broj članaka",
    color = "Kategorija"
  )
```



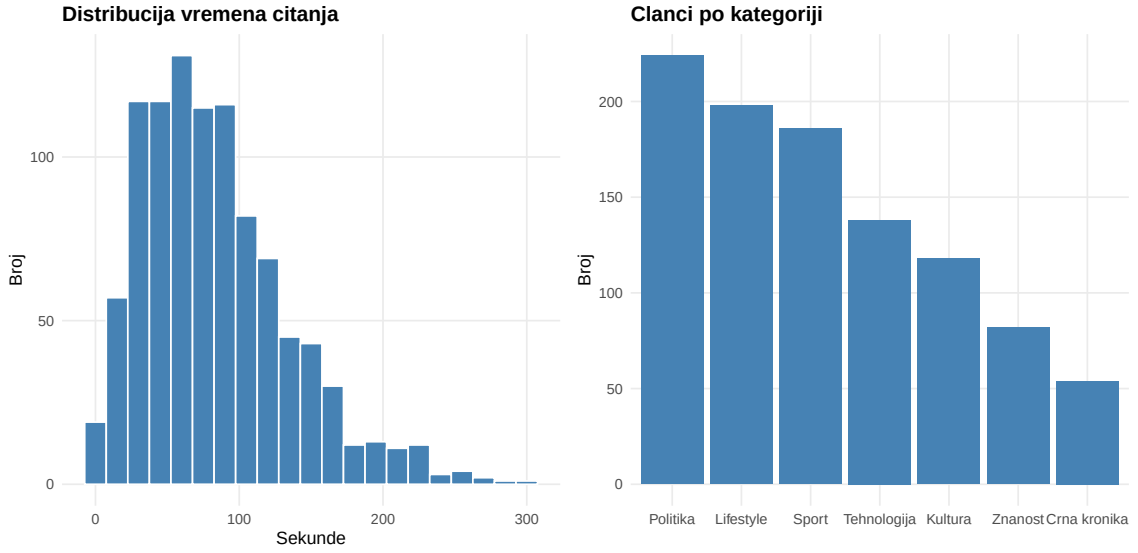
Svaka kategorija ima vlastitu liniju jer je `color = category` mapirana unutar `aes()`. Politika i sport imaju različite dnevne ritmove, što ima smisla. Sportski sadržaj se više objavljuje popodne i navečer (kad su rezultati utakmica), dok je politika koncentrirana u jutarnjim satima.

---

16 Kombiniranje grafova s patchwork

U izvještajima i radovima često trebate više grafova na jednoj stranici. Paket patchwork omogućuje elegantno slaganje ggplot2 grafova.

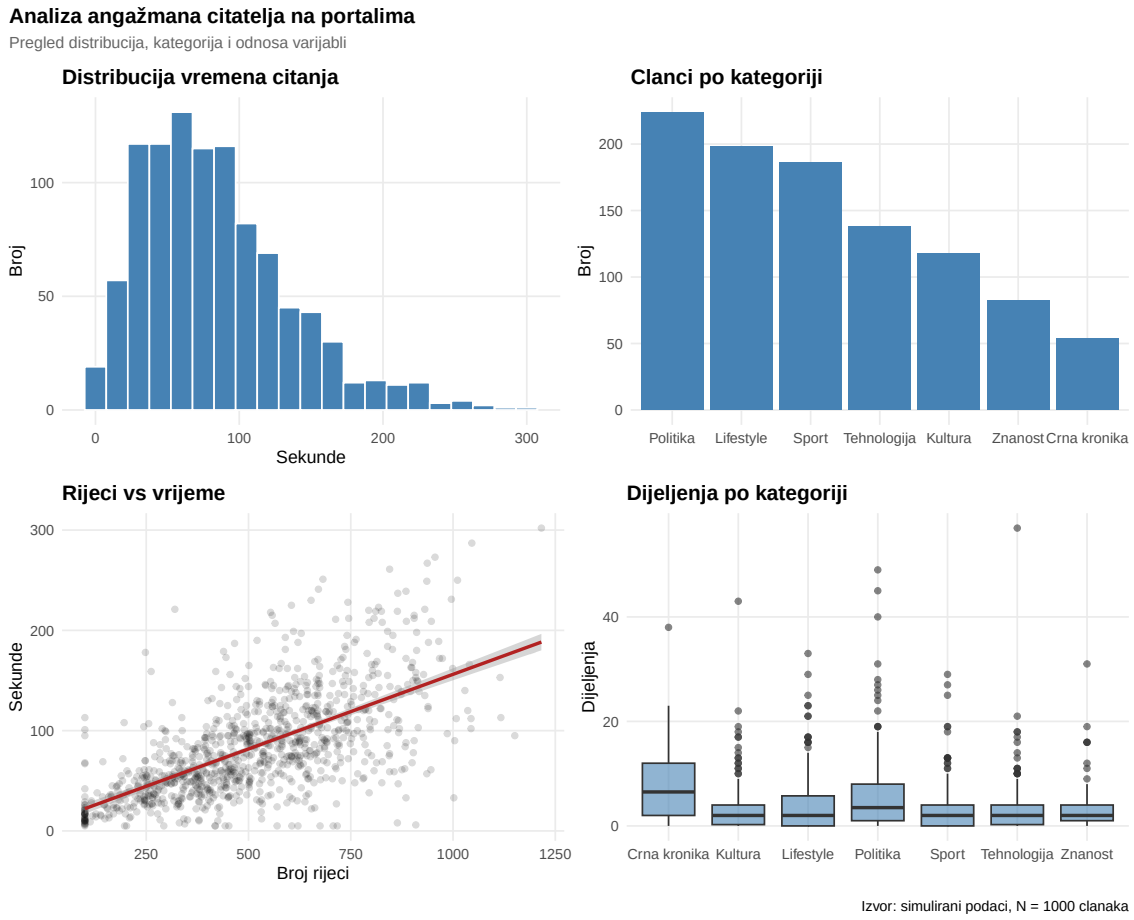
```r
library(patchwork)

p1 <- ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram(binwidth = 15, fill = "steelblue", color = "white") +
  labs(title = "Distribucija vremena čitanja", x = "Sekunde", y = "Broj")

p2 <- ggplot(clanci, aes(x = fct_infreq(category))) +
  geom_bar(fill = "steelblue") +
  labs(title = "Članci po kategoriji", x = NULL, y = "Broj")

p1 + p2
```



Operator + slaže grafove jedan do drugoga. Alternativno, / slaže vertikalno (jedan iznad drugoga), a | eksplicitno horizontalno.

```r
p3 <- ggplot(clanci, aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.15) +
  geom_smooth(method = "lm", color = "firebrick") +
  labs(title = "Riječi vs vrijeme", x = "Broj riječi", y = "Sekunde")

p4 <- ggplot(clanci, aes(x = category, y = shares)) +
  geom_boxplot(fill = "steelblue", alpha = 0.6) +
  labs(title = "Dijeljenja po kategoriji", x = NULL, y = "Dijeljenja")

(p1 | p2) / (p3 | p4) +
  plot_annotation(
    title = "Analiza angažmana čitatelja na portalima",
    subtitle = "Pregled distribucija, kategorija i odnosa varijabli",
```

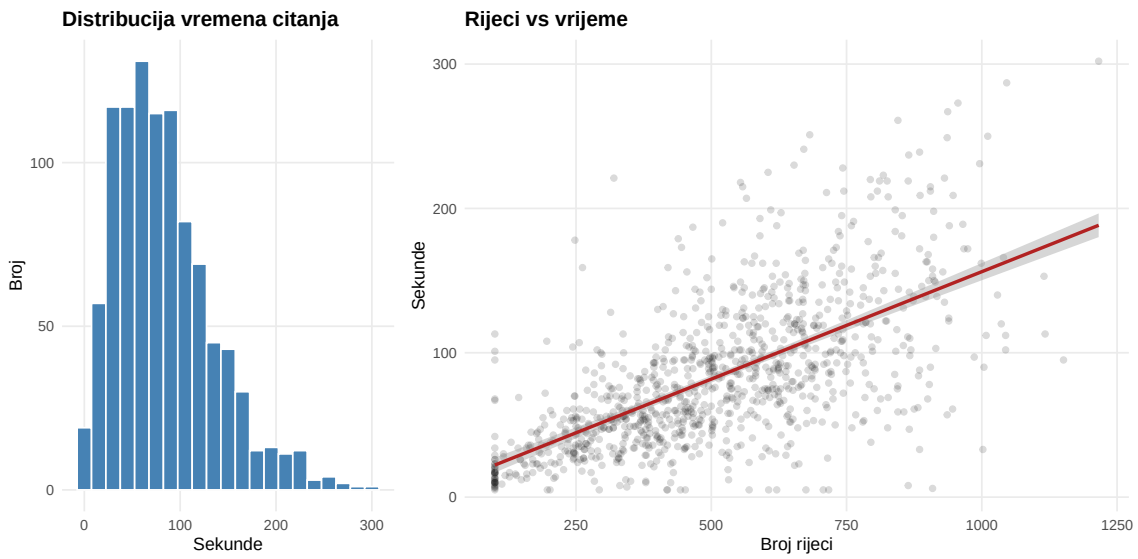
```
    caption = "Izvor: simulirani podaci, N = 1000 članaka"
  )
```



Zagrade i operatori kontroliraju raspored — `(p1 | p2) / (p3 | p4)` kreira matricu 2x2. `plot_annotation()` dodaje zajednički naslov, podnaslov i caption cijeloj kompoziciji. Ovo je profesionalan način za prezentiranje više analiza na jednom mjestu.

Patchwork podržava i `plot_layout()` za finiju kontrolu.

```r
p1 + p3 + plot_layout(widths = c(1, 2))
```



Argument `widths = c(1, 2)` daje drugom grafu dvostruku širinu — slično, `heights` kontrolira relativne visine za vertikalni raspored.

---

17 Spremanje grafova: ggsave()

Funkcija `ggsave()` sprema zadnji ggplot2 graf u datoteku. Podržava sve uobičajene formate poput PNG, PDF, SVG, JPEG i TIFF.

```r
# Spremi zadnji graf kao PNG
ggsave("angažman_portali.png", width = 10, height = 6, dpi = 300)

# Spremi specifični graf kao PDF (vektorski format, idealan za tisak)
ggsave("scatterplot.pdf", plot = p3, width = 8, height = 5)

# Spremi za prezentaciju (veće dimenzije)
ggsave("prezentacija.png", width = 12, height = 7, dpi = 150)
```

Tri ključna argumenta uključuju `width` i `height` (dimenzije u inčima) te `dpi` (rezolucija za rasterske formate). Za tisak koristite `dpi = 300`, za prezentacije `dpi = 150`, za web `dpi = 96`.

PDF format je vektorski, što znači da se skalira bez gubitka kvalitete — idealan je za akademske radove i tisak. PNG je rasterski i bolji je za web i prezentacije.

Praktični savjet

Definirajte standardne dimenzije za svoj projekt i koristite ih konzistentno. Na primjer, za Quarto dokument koji se renderira u HTML, `fig-width: 8` i `fig-height: 5` u chunk opcijama rade dobro za većinu grafova. Za prezentacije, koristite šire dimenzije (10x6). Za akademske radove, uže (6x4). Konzistentne dimenzije daju profesionalan izgled cijelom dokumentu.

---

18 Česte greške i kako ih izbjeći

Učenje ggplot2 dolazi s karakterističnim setom grešaka. Prepoznavanje najčešćih štedi sate frustracije.

18.1 Greška 1: + umjesto |> (i obrnuto)

```
# KRIVO: pipe unutar ggplot lanca
ggplot(clanci, aes(x = time_on_page)) |>
  geom_histogram()

# ISPRAVNO: + za dodavanje slojeva
ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram()
```

Unutar ggplot2 lanca koristite `+` za dodavanje slojeva. Pipe (`|>`) koristite za dplyr operacije PRIJE `ggplot()`. Tipičan obrazac je `data |> filter(...) |> ggplot(...) + geom_*()` — prelazak s pipe na plus događa se na poziv `ggplot()`.

18.2 Greška 2: kontinuirana varijabla u fill/color za bar chart

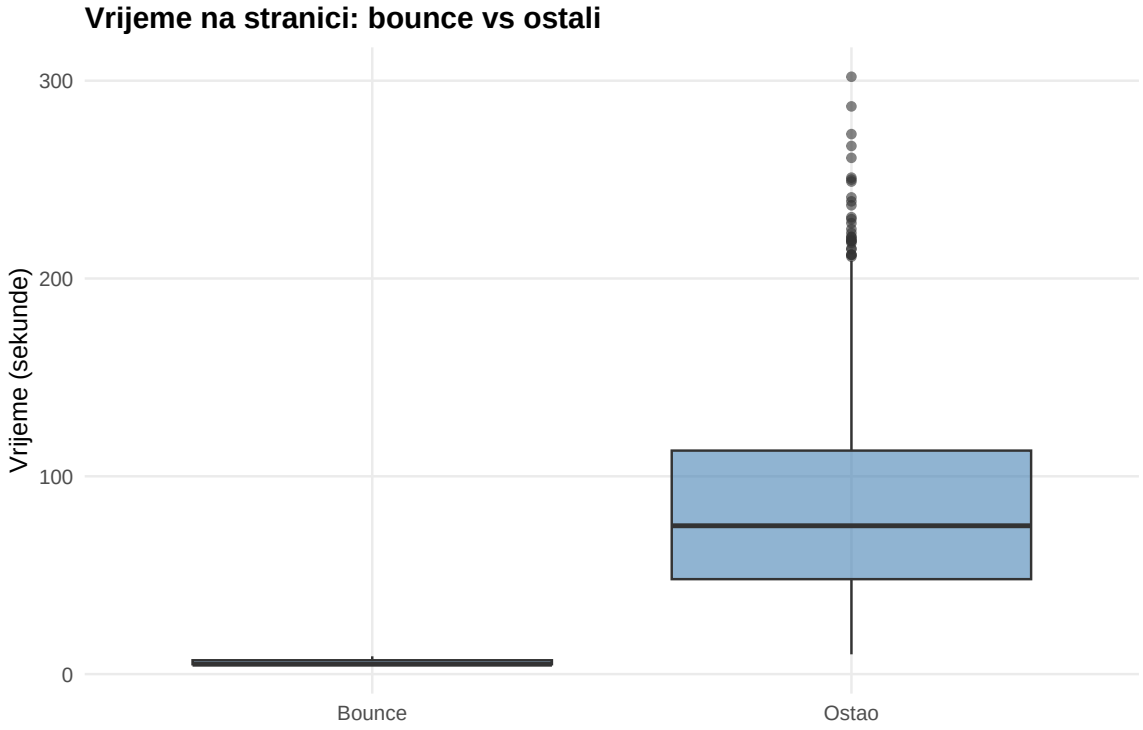
```
# ZBUNJUJUĆE: numerička varijabla kao boja u bar chartu
ggplot(clanci, aes(x = category, fill = word_count)) +
  geom_bar()

# BOLJE: kategorička varijabla za fill
ggplot(clanci, aes(x = category, fill = headline_style)) +
  geom_bar(position = "dodge")
```

18.3 Greška 3: previše informacija u jednom grafu

Ako imate sedam kategorija, četiri boje, liniju trenda, legendu i facetiranje, rezultat je vizualni kaos. Dobra vizualizacija komunicira jednu poruku jasno. Ako trebate reći više, napravite više grafova.

18.4 Greška 4: zaboravljanje na NA

```
# Logičke varijable TRUE/FALSE se ponekad pretvaraju u NA
clanci |>
  mutate(bounce_label = if_else(bounce, "Bounce", "Ostao")) |>
  ggplot(aes(x = bounce_label, y = time_on_page)) +
  geom_boxplot(fill = "steelblue", alpha = 0.6) +
  labs(
    title = "Vrijeme na stranici: bounce vs ostali",
    x = NULL,
    y = "Vrijeme (sekunde)"
  )
```



Ako u podatcima postoji NA u varijabli koja definira grupu, ggplot će napraviti zaseban panel ili stupac za NA. Uvijek provjerite podatke prije vizualizacije i odlučite želite li NA prikazati, filtrirati ili rekodirati.

19 Kompletna analiza: od pitanja do gotovog grafa

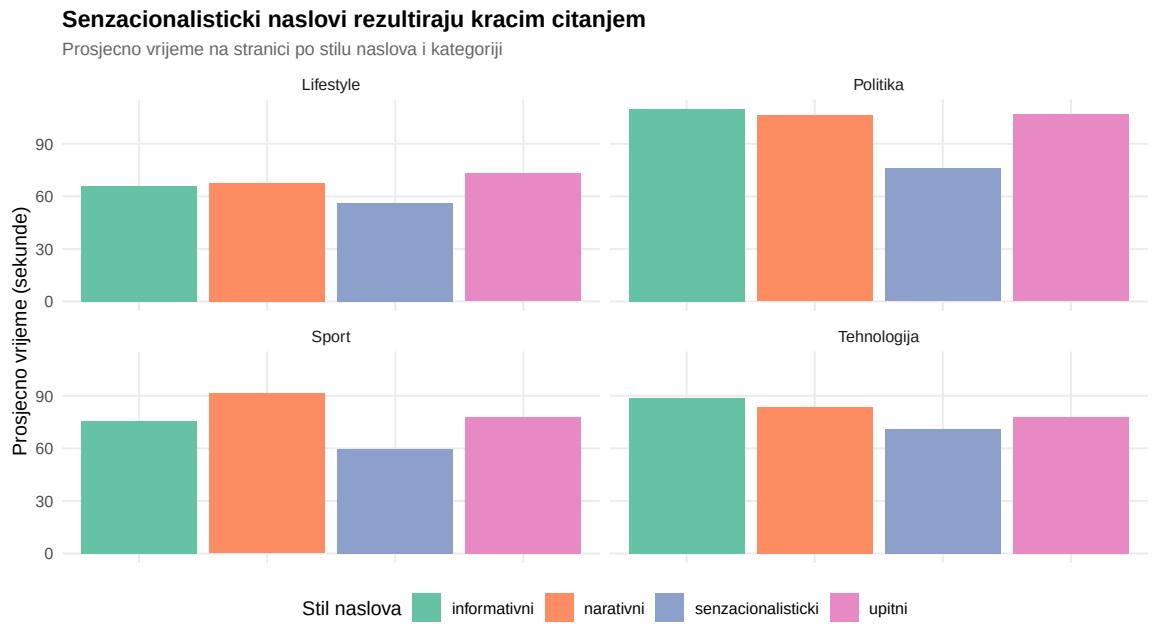
Zaokružimo predavanje kompletnim primjerom koji prolazi sve korake — definiranje pitanja, priprema podataka, odabir grafa, izgradnja i poliranje.

Pitanje — Kako se angažman čitatelja (vrijeme čitanja i dijeljenje) razlikuje ovisno o stilu naslova i kategoriji članka?

```
# Priprema podataka
angazman <- clanci |>
  filter(category %in% c("Politika", "Sport", "Tehnologija", "Lifestyle")) |>
  group_by(category, headline_style) |>
  summarise(
    n = n(),
    prosjek_vrijeme = mean(time_on_page),
    prosjek_dijeljenja = mean(shares),
    udio_bounce = mean(bounce),
    .groups = "drop"
  ) |>
  filter(n >= 5)

angazman
```

```
# A tibble: 16 x 6
   category  headline_style     n prosjek_vrijeme prosjek_dijeljenja udio_bounce
   <chr>     <chr>          <int>           <dbl>              <dbl>       <dbl>
 1 Lifestyle informativni      71            66.0               1.87      0.0141
 2 Lifestyle narativni         39            67.5               1.79      0.0513
 3 Lifestyle senzacionalis~    40            56.1               9         0.05
 4 Lifestyle upitni            48            73.0               5.77      0
 5 Politika  informativni      88           110.                3.43      0
 6 Politika  narativni         40           106.                3.62      0.025
 7 Politika  senzacionalis~    54            75.9              11.6       0.0185
 8 Politika  upitni            42           107.                6.36      0
 9 Sport     informativni      62            75.5               1.94      0
10 Sport     narativni         38            91.3               1.68      0
11 Sport     senzacionalis~    41            59.6               6.24      0.0488
12 Sport     upitni            45            77.7               4.13      0
13 Tehnolog~ informativni      47            89.0               1.04      0.0638
14 Tehnolog~ narativni         24            83.5               2.21      0
15 Tehnolog~ senzacionalis~    28            70.9               8.21      0.0357
16 Tehnolog~ upitni            39            77.7               4.59      0.0513
```

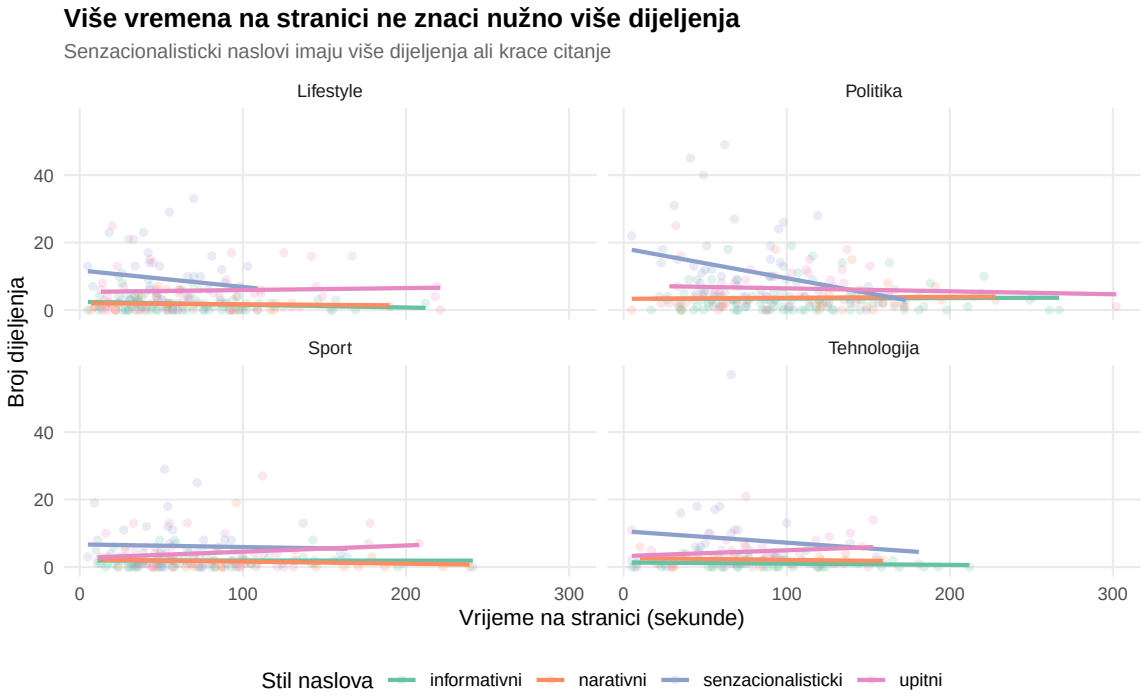
```
# Graf 1: Prosječno vrijeme čitanja po stilu naslova i kategoriji
angazman |>
  ggplot(aes(x = headline_style, y = prosjek_vrijeme, fill = headline_style)) +
  geom_col() +
  facet_wrap(~category) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "Senzacionalistički naslovi rezultiraju kraćim čitanjem",
    subtitle = "Prosječno vrijeme na stranici po stilu naslova i kategoriji",
    x = NULL,
    y = "Prosječno vrijeme (sekunde)",
    fill = "Stil naslova"
  ) +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    legend.position = "bottom"
  )
```



Uklonili smo oznake na x osi (`element_blank()`) jer legenda na dnu sadrži istu informaciju. Ovo smanjuje vizualni šum i čini graf čitljivijim.

```
# Graf 2: Odnos vremena i dijeljenja, po stilu naslova
clanci |>
  filter(category %in% c("Politika", "Sport", "Tehnologija", "Lifestyle")) |>
  ggplot(aes(x = time_on_page, y = shares, color = headline_style)) +
  geom_point(alpha = 0.2) +
```

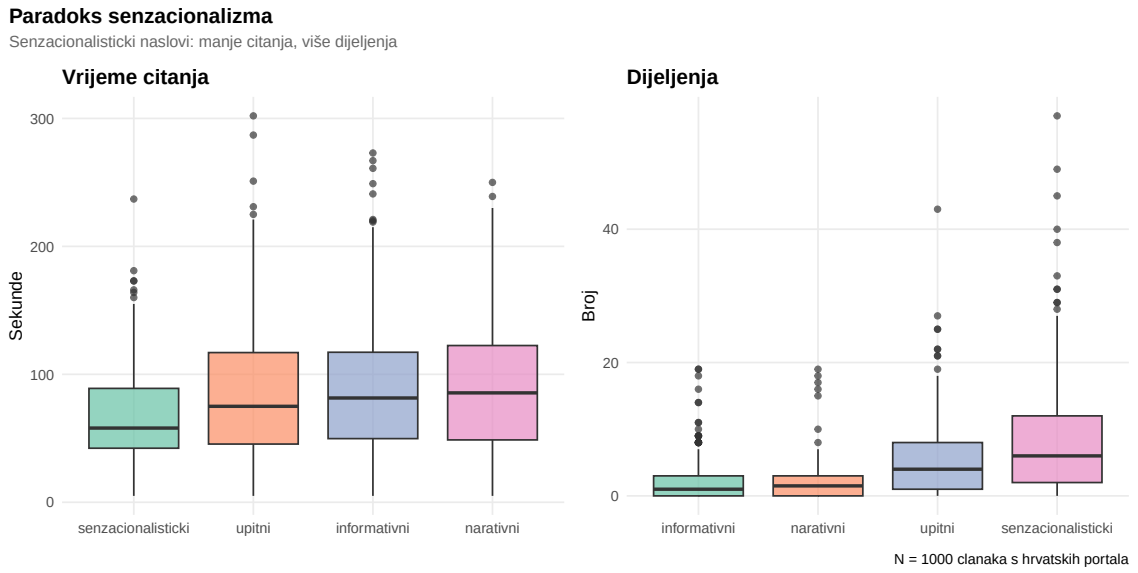
```
  geom_smooth(method = "lm", se = FALSE) +
  scale_color_brewer(palette = "Set2") +
  facet_wrap(~category) +
  labs(
    title = "Više vremena na stranici ne znači nužno više dijeljenja",
    subtitle = "Senzacionalistički naslovi imaju više dijeljenja ali kraće čitanje",
    x = "Vrijeme na stranici (sekunde)",
    y = "Broj dijeljenja",
    color = "Stil naslova"
  ) +
  theme(legend.position = "bottom")
```



Ovaj graf otkriva zanimljiv paradoks. Senzacionalistički naslovi privlače klikove i dijeljenja, ali čitatelji provode manje vremena na članku. Informativni i narativni naslovi imaju manje dijeljenja ali duže čitanje. Ovo je klasična dilema digitalnog novinarstva — optimizirate li za klikove ili za dubinski angažman?

```
# Graf 3: Kompozitni prikaz s patchwork
graf_a <- clanci |>
  mutate(headline_style = fct_reorder(headline_style, time_on_page)) |>
  ggplot(aes(x = headline_style, y = time_on_page, fill = headline_style)) +
  geom_boxplot(alpha = 0.7, show.legend = FALSE) +
  scale_fill_brewer(palette = "Set2") +
  labs(title = "Vrijeme čitanja", x = NULL, y = "Sekunde")
```

```r
graf_b <- clanci |>
  mutate(headline_style = fct_reorder(headline_style, shares)) |>
  ggplot(aes(x = headline_style, y = shares, fill = headline_style)) +
  geom_boxplot(alpha = 0.7, show.legend = FALSE) +
  scale_fill_brewer(palette = "Set2") +
  labs(title = "Dijeljenja", x = NULL, y = "Broj")

graf_a + graf_b +
  plot_annotation(
    title = "Paradoks senzacionalizma",
    subtitle = "Senzacionalistički naslovi: manje čitanja, više dijeljenja",
    caption = "N = 1000 članaka s hrvatskih portala"
  )
```



Tri grafa zajedno ispričali su kompletnu priču. Od sažetka podataka do usmjerenog nalaza, svaki graf ima jasnu poruku. Ovo je razina vizualizacije koja se očekuje u akademskim radovima, poslovnim izvještajima i novinarskim analizama.

! Ključni zaključci

1. ggplot2 graf se gradi od tri obavezne komponente: podaci, estetike (`aes()`) i geometrija (`geom_*()`). Sve ostalo (skale, faceti, teme) je opcionalno ali važno za profesionalan izgled.

2. Estetike unutar `aes()` mapiraju varijable na vizualna svojstva (i kreiraju legendu). Estetike izvan `aes()` postavljaju fiksne vrijednosti. Miješanje ova dva pristupa je

najčešći izvor zbunjenosti.

3. Histogram i density prikazuju distribuciju jedne varijable. Bar chart prikazuje kategorije. Boxplot uspoređuje distribucije između grupa. Scatterplot prikazuje odnos dviju varijabli. Odabir grafa ovisi o tipovima varijabli koje imate.

4. Facetiranje (`facet_wrap()`, `facet_grid()`) dijeli graf na panele po grupama. Gotovo uvijek je čitljivije od preklapanja mnogo grupa u jednom grafu.

5. Teme kontroliraju vizualne elemente koji nisu podaci. `theme_minimal()` i `theme_bw()` su dobri izbori za profesionalan rad. `theme_set()` postavlja globalnu temu za cijeli dokument.

6. Boje se biraju ovisno o tipu podataka: kvalitativne palete (Set2, Dark2) za kategorije, sekvencijalne (Blues, viridis) za kontinuirane varijable. Viridis palete su pristupačne osobama s poremećajem vida boja.

7. `labs()` je obavezna funkcija za svaki graf. Formulirajte naslov kao nalaz, ne kao opis. Dodajte `caption` za izvor podataka.

8. `ggsave()` sprema grafove u datoteku. PDF za tisak (vektorski), PNG za web (rasterski). Koristite `dpi = 300` za tisak.

9. Patchwork kombinira više grafova u jednu kompoziciju operatorima `+`, `/`, `|`. `plot_annotation()` dodaje zajednički naslov.

10. Linijski grafovi (`geom_line()`) su prirodan izbor za podatke s redoslijedom, posebno vremenske serije.

11. Unutar ggplot2 lanca koristite `+` za slojeve. Pipe (`|>`) koristite za dplyr PRIJE `ggplot()`. Prelazak je na poziv `ggplot()`.

12. Dobra vizualizacija komunicira jednu poruku jasno. Ako trebate reći više, napravite više grafova. Vizualni kaos s previše slojeva je gori od praznog platna.

⚠ Priprema za sljedeći tjedan

Sljedeći tjedan bavimo se **programiranjem u R-u**. Naučit ćete pisanje funkcija, uvjetne naredbe, petlje i organizaciju ponovljivih analiza. Fokus nije na tome da postanete programeri, nego na tome da napišete čist, ponovljiv kod koji možete pokrenuti ponovno kad dobijete nove podatke.
Za pripremu:

1. Ponovite sve tipove grafova iz ovog predavanja. Za svaki pokušajte promijeniti barem jedan argument i vidjeti što se događa.
2. Napravite tri grafa iz podataka `article_engagement.csv` koji odgovaraju na

sljedeće pitanje — razlikuju li se portali po angažmanu čitatelja? Koristite barem jedan histogram, jedan boxplot i jedan bar chart.
3. Kombinirajte ta tri grafa pomoću patchwork u jednu kompoziciju s zajedničkim naslovom.
4. Pročitajte poglavlje 8 iz Navarro (Learning Statistics with R) o osnovama programiranja.

---

20 Dodatno čitanje

Obavezno

Wickham, H. & Grolemund, G. (2023). *R for Data Science* (2nd edition), Chapters 2, 10, 11 i 12. Besplatno dostupno na r4ds.hadley.nz. Poglavlje 2 daje brzi uvod u vizualizaciju, poglavlje 10 pokriva EDA, poglavlja 11 i 12 detaljno obrađuju komunikaciju putem grafova i slojeve ggplot2.

Navarro, D. (2018). *Learning Statistics with R*, Chapter 6. Besplatno dostupno na learningstatisticswithr.com. Poglavlje koristi base R grafiku, ali koncepti izbora grafa su univerzalni.

Preporučeno

Healy, K. (2018). *Data Visualization: A Practical Introduction*. Princeton University Press. Besplatno dostupno na socviz.co. Izvrsna knjiga o ggplot2 s naglaskom na principe vizualizacije u društvenim znanostima.

Wilke, C. O. (2019). *Fundamentals of Data Visualization*. O'Reilly. Besplatno dostupno na clauswilke.com/dataviz. Fokus na principima vizualizacije neovisno o alatu.

Scherer, C. (2022). *A ggplot2 Tutorial for Beautiful Plotting in R*. Besplatno dostupno na cedricscherer.netlify.app/2019/08/05/a-ggplot2-tutorial-for-beautiful-plotting-in-r. Detaljan vodič za profesionalno poliranje grafova u ggplot2.

---

21 Pojmovnik

Pojam	Objašnjenje
ggplot2	R paket za vizualizaciju podataka temeljen na gramatici grafike. Dio tidyverse ekosustava.

Pojam	Objašnjenje
Gramatika grafike	Sustav komponenti (podaci, estetike, geometrija, skale, faceti, teme) koje se kombiniraju u slojeve za kreiranje grafova.
`aes()`	Funkcija za mapiranje varijabli na vizualne dimenzije grafa (x os, y os, boja, veličina, oblik).
Geometrija (`geom_*()`)	Vizualni oblik za prikaz podataka. Svaki tip grafa ima svoju geom funkciju.
`geom_histogram()`	Geometrija za histogram. Argument `binwidth` kontrolira širinu bina.
`geom_density()`	Geometrija za graf gustoće distribucije.
`geom_bar()`	Geometrija za bar chart koji automatski broji opažanja po kategorijama.
`geom_col()`	Geometrija za bar chart s prethodno izračunatim y vrijednostima.
`geom_boxplot()`	Geometrija za boxplot koji prikazuje medijan, kvartile i outliere.
`geom_violin()`	Geometrija za violin plot koji prikazuje oblik distribucije.
`geom_point()`	Geometrija za scatterplot.
`geom_jitter()`	Varijanta `geom_point()` s nasumičnim pomakom za izbjegavanje preklapanja.
`geom_smooth()`	Geometrija za liniju trenda. Default LOESS, `method = "lm"` za linearnu.
`geom_line()`	Geometrija za linijski graf. Pogodna za vremenske serije i podatke s redoslijedom.
`facet_wrap()`	Dijeli graf na panele po jednoj varijabli. Argumenti: `ncol`, `scales`.
`facet_grid()`	Dijeli graf na matricu panela po dvjema varijablama. Sintaksa: `retci ~ stupci`.
`labs()`	Funkcija za naslove, podnaslove, oznake osi, legende i caption.
`theme_minimal()`	Ugrađena tema: čista, bez okvira, minimalna mreža. Popularna za profesionalni rad.
`theme_bw()`	Ugrađena tema: bijela pozadina s crnim okvirom.
`theme()`	Funkcija za detaljnu prilagodbu vizualnih elemenata (fontovi, margine, legenda, mreža).
`theme_set()`	Postavlja globalnu temu za sve grafove u dokumentu.
`element_text()`	Unutar `theme()`: kontrolira svojstva teksta (veličina, bold, boja, kut).

Pojam	Objašnjenje
`element_blank()`	Unutar `theme()`: potpuno uklanja element (mreža, oznake, rubovi).
`scale_color_manual()`	Ručni odabir boja za `color` estetiku.
`scale_fill_manual()`	Ručni odabir boja za `fill` estetiku.
`scale_color_brewer()`	ColorBrewer palete za `color`. Tipovi: kvalitativne, sekvencijalne, divergentne.
`scale_fill_brewer()`	ColorBrewer palete za `fill`.
`scale_color_viridis_c()`	Viridis paleta za kontinuirane varijable. Pristupačna za poremećaj vida boja.
`scale_color_viridis_d()`	Viridis paleta za diskretne varijable.
`alpha`	Estetika za transparentnost. Od 0 (prozirno) do 1 (neprozirno).
`fill`	Estetika za boju ispune (stupci, pravokutnici, područja).
`color`	Estetika za boju ruba ili linije (točke, linije, rubovi).
`position = "dodge"`	Stupci jedne do drugih u grupiranom bar chartu.
`position = "fill"`	Normalizira stupce na proporcije.
`fct_infreq()`	Sortira faktor po frekvenciji.
`fct_reorder()`	Sortira faktor po vrijednostima druge varijable.
`coord_flip()`	Zamjenjuje x i y os za horizontalne grafove.
`coord_cartesian()`	Zumira graf bez uklanjanja podataka.
`ggsave()`	Sprema graf u datoteku. Argumenti: širina, visina, dpi, format.
patchwork	Paket za kombiniranje više ggplot2 grafova. Operatori: `+` (horizontalno), `/` (vertikalno), `|` (horizontalno).
`plot_annotation()`	Patchwork funkcija za zajednički naslov kompozicije.
Whisker	Linije iz boxplota do 1.5 x IQR od kvartila.
Outlier	Točka udaljena više od 1.5 x IQR od kvartila.
DPI	Dots per inch. Rezolucija rasterske slike. 300 za tisak, 150 za prezentacije, 96 za web.